



**UNIVERSIDAD NACIONAL DEL ALTIPLANO DE PUNO**  
**FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**  
**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**



**“SISTEMA DE INFORMACIÓN CON ARQUITECTURA MVC PARA LA  
RECAUDACIÓN TRIBUTARIA POR OBRAS PÚBLICAS DE PISTAS Y  
VEREDAS DE LA MUNICIPALIDAD PROVINCIAL DE PUNO”**

**TESIS**

**PRESENTADA POR:**

**Bach. JUAN CARLOS HUANCA MOLLINEDO**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

**INGENIERO ESTADÍSTICO E INFORMÁTICO**

**PUNO – PERÚ**

**2020**



## DEDICATORIA

A mi querida madre Rosa Yolanda por su apoyo incondicional, que pese a difíciles momentos de la vida demostró valentía y esfuerzo para salir adelante.



## AGRADECIMIENTOS

En especial a la Universidad Nacional del Altiplano de Puno y a la Facultad de Ingeniería Estadística e Informática, la cual me abrió las puertas para formarme profesionalmente.

A mis profesores quienes me incentivaron a seguir adelante y sin su apoyo esto no hubiera sido posible.

Y a todas aquellas personas que siempre estuvieron a mi lado apoyándome en las buenas y en las malas.



# ÍNDICE GENERAL

**DEDICATORIA**

**AGRADECIMIENTOS**

**ÍNDICE GENERAL**

**ÍNDICE DE FIGURAS**

**ÍNDICE DE TABLAS**

**ÍNDICE DE ACRÓNIMOS**

**RESUMEN ..... 11**

**ABSTRACT..... 12**

## **CAPÍTULO I**

### **INTRODUCCIÓN**

**1.1. PLANTEAMIENTO DEL PROBLEMA ..... 14**

**1.2. FORMULACIÓN DEL PROBLEMA..... 15**

**1.3. HIPÓTESIS DE LA INVESTIGACIÓN..... 15**

**1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN..... 15**

**1.5. OBJETIVOS DE LA INVESTIGACIÓN ..... 16**

1.5.1. Objetivo General..... 16

1.5.2. Objetivos Específicos ..... 16

## **CAPÍTULO II**

### **REVISIÓN DE LITERATURA**

**2.1. MARCO TEÓRICO..... 17**

2.1.1. Antecedentes De La Investigación..... 17

**2.2. MARCO CONCEPTUAL..... 20**

2.2.1. Ley De Tributación Municipal..... 20

2.2.2. Sistema de Información ..... 21

2.2.3. Modelo Vista Controlador (MVC)..... 22



2.2.4. Lenguaje Unificado de Modelado (UML) .....	24
2.2.5. MariaDB .....	24
2.2.6. HTML5 .....	24
2.2.7. PHP .....	24
2.2.8. JavaScript.....	25
2.2.9. AJAX .....	25
2.2.10. NGINX.....	26
2.2.11. CodeIgniter .....	27

### **CAPÍTULO III**

#### **MATERIALES Y MÉTODOS**

<b>3.1. POBLACIÓN Y MUESTRA .....</b>	<b>28</b>
<b>3.2. TIPO DE INVESTIGACIÓN .....</b>	<b>29</b>
<b>3.3. DISEÑO DE LA INVESTIGACIÓN.....</b>	<b>30</b>
<b>3.4. MÉTODO DE VALIDACIÓN DE SOFTWARE.....</b>	<b>30</b>
<b>3.5. MÉTODO DE RECOPIACIÓN DE DATOS .....</b>	<b>31</b>
<b>3.6. MÉTODO DE TRATAMIENTO DE DATOS .....</b>	<b>32</b>
<b>3.7. MATERIAL EXPERIMENTAL .....</b>	<b>35</b>
3.7.1. Hardware.....	35
3.7.2. Software .....	35
3.7.3. Material De Escritorio.....	36
<b>3.8. METODOLOGÍA DE DESARROLLO DEL SISTEMA.....</b>	<b>36</b>
3.8.1. Metodología de desarrollo de software (XP) .....	36

### **CAPÍTULO IV**

#### **RESULTADOS Y DISCUSIÓN**

<b>4.1. ANÁLISIS DEL SISTEMA .....</b>	<b>39</b>
4.1.1. Análisis de Requerimientos .....	39
4.1.2. Iteraciones .....	48



4.1.3. Requerimientos no funcionales.....	49
<b>4.2. DISEÑO DEL SISTEMA.....</b>	<b>50</b>
4.2.1. Diagrama de Casos De Uso .....	50
4.2.2. Diagrama de Secuencia.....	51
4.2.3. Diagrama de Clases.....	55
4.2.4. Diagrama de Actividades.....	56
4.2.5. Diagrama de Componentes .....	59
4.2.6. Diseño de la base de datos .....	59
4.2.7. Arquitectura Física del Sistema .....	60
4.2.8. Arquitectura Lógica del Sistema.....	61
<b>4.3. CODIFICACIÓN .....</b>	<b>61</b>
<b>4.4. MÓDULOS DEL SOFTWARE .....</b>	<b>61</b>
<b>4.5. PRUEBA DE SOFTWARE .....</b>	<b>62</b>
4.5.1. Validación Del Sistema.....	62
<b>4.6. PRUEBA DE HIPÓTESIS.....</b>	<b>63</b>
<b>4.7. DISCUSIÓN.....</b>	<b>66</b>
<b>V. CONCLUSIONES.....</b>	<b>68</b>
<b>VI. RECOMENDACIONES .....</b>	<b>70</b>
<b>VII. REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>71</b>
<b>ANEXOS.....</b>	<b>74</b>
Anexo A. Encuesta sobre la implementación del sistema de información mejora la gestión administrativa de recaudación tributaria por obras públicas. ....	75
Anexo B. Encuesta a los trabajadores y funcionamiento de SIRPACEPV .....	76
Anexo C. Encuesta antes y después de implementar SIRPACEPV.....	77
Anexo D. Normalización de Base de Datos. ....	78
Anexo E. Instalación y configuración de CodeIgniter .....	81
Anexo F. Estructura del código fuente basado en Modelo Vista Controlador. ....	83



Anexo G. Resultado de la implementación del sistema de información .....	93
Anexo H. Fichas de evaluación de ISO – 9126.....	99
Anexo I. Datos del tiempo (en minutos) de Demora en la Atención Antes(y) y Después (x) de la implementación del sistema.....	100
Anexo J. Resultados del cuestionario acerca de “SIRPACEPV”.....	101
Anexo K. Manual de Usuario “SIRPACEPV”.....	106

**Área:** Informática

**Línea:** Base De Datos y Sistemas De Información

**FECHA DE SUSTENTACIÓN:** 23 de diciembre del 2020



## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Modelo Vista Controlador (MVC).....	23
<b>Figura 2.</b> Modelo de Aplicación Web Ajax.....	26
<b>Figura 3.</b> Pre Prueba y Post Prueba .....	30
<b>Figura 4.</b> Modelo de Calidad del Software ISO – 9126 .....	31
<b>Figura 5.</b> Proceso XP .....	37
<b>Figura 6.</b> Diagrama de Casos De Uso.....	50
<b>Figura 7.</b> Diagrama de Secuencia - Acceso al Sistema .....	52
<b>Figura 8.</b> Diagrama de Secuencia - Registrar Contribuyente y Generar Monto.....	53
<b>Figura 9.</b> Diagrama de secuencia - Registro de Pagos.....	54
<b>Figura 10.</b> Diagrama de secuencia - Consulta de pagos realizados.....	55
<b>Figura 11.</b> Diagrama de Clases.....	56
<b>Figura 12.</b> Diagrama de Actividad - Registro de Contribuyente.....	57
<b>Figura 13.</b> Diagrama de Actividad - Registrar Pago.....	58
<b>Figura 14.</b> Diagrama de Actividad - Consultar Pagos .....	58
<b>Figura 15.</b> Diagrama de componentes del Sistema de Información .....	59
<b>Figura 16.</b> Diseño de Base de Datos.....	60
<b>Figura 17.</b> Arquitectura Física del Sistema.....	60
<b>Figura 18.</b> Arquitectura Lógica del Sistema.....	61
<b>Figura 19.</b> Implementación de "SIRPACEPV" mejora la Gestión Administrativa.....	65
<b>Figura 20.</b> Tiempo de atención con el uso del sistema "SIRPACEPV" .....	66





## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Historia de Usuario - Autenticación de Usuario .....	40
<b>Tabla 2.</b> Historia de Usuario - Registrar Obras.....	41
<b>Tabla 3.</b> Historia de Usuario - Registrar Contribuyente .....	41
<b>Tabla 4.</b> Historia de Usuario - Generar Monto de Contribución .....	42
<b>Tabla 5.</b> Historia de Usuario - Registrar pagos.....	44
<b>Tabla 6.</b> Historia de Usuario - Generar Constancias de cancelación y no adeudo. ....	45
<b>Tabla 7.</b> Historia de Usuario - Generar Notificaciones.....	46
<b>Tabla 8.</b> Historia de Usuario - Visualizar archivos generados.....	47
<b>Tabla 9.</b> Historia de Usuario - Ver y generar Reporte de pagos .....	48
<b>Tabla 10.</b> Tabla de ilustraciones .....	49
<b>Tabla 11.</b> Medición de calidad del software estándar ISO - 9126 .....	62
<b>Tabla 12.</b> La implementación del sistema “SIRPACEPV” mejora la gestión administrativa .....	65
<b>Tabla 13.</b> Tiempo de atención con el uso del sistema “SIRPACEPV”. .....	66



## ÍNDICE DE ACRÓNIMOS

<b>AJAX</b>	: Asynchronous JavaScript and XML.
<b>CSS</b>	: Hojas de Estilo en Cascada.
<b>PHP</b>	: Pre-procesador de hiper-texto.
<b>HTML</b>	: Lenguaje de marcas de hipertexto.
<b>ISO</b>	: Organización de Estándares Internacionales.
<b>MySQL</b>	: Lenguaje de consulta estructurado.
<b>MVC</b>	: Modelo Vista Controlador.
<b>UML</b>	: Lenguaje de modelado unificado.
<b>SGBD</b>	: Sistema de gestión de base de datos.
<b>XP</b>	: Programación Extrema.
<b>SIRPACEPV</b>	: Sistema de Registro de Pagos por contribución Especial de Pistas y Veredas.



## RESUMEN

La finalidad de esta investigación es implementar un Sistema de Información que permita optimizar y modernizar la gestión administrativa de recaudación tributaria por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno, partiendo de actividades básicas, con el fin de minimizar el tiempo de atención a los contribuyentes. Para el desarrollo del software se utilizó la metodología ágil extreme programming(XP), por su mayor afinidad, claridad de actividades y el coraje para enfrentar los cambios, esta metodología se basa en la retroalimentación continua y comunicación con el usuario final, requisito que permitió construir un sistema de información de fácil comunicación con los operadores. Para el modelado de datos se usó UML (Lenguaje Unificado de Modelado) y como patrón de arquitectura de software se aplicó MVC (Modelo Vista Controlador) que permitió clasificar del código en tres capas diferentes, este patrón favorece que el sistema pueda adaptarse a cambios en el futuro. Para la validación del sistema se aplicó una ficha de evaluación de calidad de producto de software del estándar ISO-9126 a los operadores del sistema. Por lo que se concluye que el Sistema de Información cumple con los requisitos planteados con un promedio de “86” puntos según la escala de calificación del estándar ISO-9126. Asimismo, se prueba que con la Implementación del Sistema de Información se reduce notablemente el tiempo de atención a los contribuyentes, dado que la prueba de hipótesis resultó significativa, donde el valor calculado es mayor al valor crítico ( $Z_c (8.99) > Z_t (1.645)$ ) con un nivel de confianza del 95%.

**Palabras Claves:** Recaudación Tributaria, Sistema de información, Implementación, Automatización, Modelo Vista Controlador.



## ABSTRACT

The purpose of this research is to implement an Information System that allows optimizing and modernizing the administrative management of tax collection for public works of tracks and sidewalks of the Municipalidad Provincial de Puno, based on basic activities, in order to minimize service time to taxpayers.

For the development of the software, the agile extreme programming (XP) methodology was used, due to its greater affinity, clarity of activities and the courage to face changes, this methodology is based on continuous feedback and communication with the end user, a requirement that allowed build an information system for easy communication with operators. UML (Unified Modeling Language) was used for data modeling and MVC (Controller View Model) was applied as a software architecture pattern that allowed classifying the code into three different layers, this pattern favors that the system can adapt to changes in the future. For the validation of the system, a software product quality evaluation sheet of the ISO-9126 standard was applied to the system operators. Therefore, it is concluded that the Information System complies with the requirements established with an average of "86" points according to the ISO-9126 rating scale. Likewise, it is proved that with the implementation of the Information System, the time of attention to taxpayers is significantly reduced, since the hypothesis test was significant, where the calculated value is bigger than the critical value ( $Z_c (8.99) > Z_t (1.645)$ ) with a confidence level of 95%.

**Keywords:** Tributary Collection, Information System, Implementation, Automation, Model View Controller.



# CAPÍTULO I

## INTRODUCCIÓN

El mundo de hoy ha generado muchos cambios, los cuales están sujetos a la manipulación de información en grandes cantidades, es así como las empresas, instituciones grandes, medianas y pequeñas hacen uso de un sistema de información para tener control de su información en cualquier momento. Lo que vale destacar son los beneficios que la tecnología informática ha provisto a la humanidad, en facilitar el desarrollo de un cálculo en corto tiempo.

Hoy en día el eficiente seguimiento y control de la información representa un reto en las instituciones en donde se manipula información, y en la mayoría de los casos el registro de hace de manera manual y una vez archivada la información representa cierto grado de dificultad para extraer en torno a un documento o información en específico. En la actualidad la Sub Gerencia de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno, es la oficina encargada de recaudar por contribución especial por obras públicas de pistas y veredas y para ello se registra los bienes inmuebles, propietarios y todos los pagos realizados por los contribuyentes, todas esas actividades se realizaban de manera manual el cual no brinda la posibilidad de modificar, actualizar en un corto tiempo, generando carga laboral para los trabajadores; otro de los servicios que es importante es la consulta de pagos realizados que se realiza manualmente, lo cual no brinda una atención de manera eficiente y oportuna a los contribuyentes para la demanda de información.

En este documento se muestra el aporte profesional mediante el Desarrollo e Implementación de un Sistema de Información para la recaudación tributaria por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno, el cual permite a los trabajadores, realizar tareas específicas como registro de información del inmueble del



contribuyente, pagos realizados y reportes en general de forma automatizada y rápida, optimizándose en gran parte la atención a los contribuyentes; se utilizó la metodología XP y el lenguaje de modelado UML que permite dar una vista detallada para el desarrollo del sistema de información.

## **1.1. PLANTEAMIENTO DEL PROBLEMA**

La Sub Gerencia de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno, es la encargada de la gestión tributaria de contribución especial de obras públicas y viene estableciendo cobros por contribución conforme lo indica el texto del artículo 62° de la Ley de Tributación Municipal, llevando sus procesos de registro, control de pagos en un archivo de hoja de cálculo (Excel) y de forma manual. Sin embargo, el proceso de recaudación es deficiente con problemas de pérdida de información, el mal aseguramiento de los datos y por la manera tradicional que se administra la información de los contribuyentes.

La falta de un Sistema de Información para la gestión administrativa de recaudación tributaria por obras públicas de pistas y veredas, ha generado una serie de desconformidades al personal que labora y a los contribuyentes; debido al tiempo empleado para la generación y extracción de información ya que todo ese procedimiento se hace de una forma no óptima, ocasionando errores y pérdida de materiales (Hojas, tinta), motivo por el cual, es necesario implantar nuevas alternativas más eficientes para optimizar tareas específicas y mejorar la atención a los contribuyentes.



## **1.2. FORMULACIÓN DEL PROBLEMA**

¿De qué manera el diseño e implementación de un Sistema de Información para la recaudación tributaria por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno, mejorará la atención a los contribuyentes?

## **1.3. HIPÓTESIS DE LA INVESTIGACIÓN**

La implementación del Sistema de Información, para la oficina de “Promoción Social y Participación Ciudadana” de la Municipalidad Provincial de Puno, reducirá el tiempo de atención de la gestión administrativa de recaudación tributaria por Obras Públicas de pistas y veredas.

## **1.4. JUSTIFICACIÓN DE LA INVESTIGACIÓN**

La Municipalidad depende en su totalidad de los ingresos que existan dentro de ésta, tanto de los otorgados por el gobierno central o como los recaudados directamente por la Municipalidad, sin embargo, tiene la doble función, la de recaudar los tributos y retornarlos a la Provincia en Obras.

Por tal razón se realizó el actual trabajo de investigación, que busca optimizar el proceso de recaudación tributaria por obras públicas de pistas y veredas mediante la implementación de un sistema de informático, al mejorar el proceso administrativo se logra recaudar mejor las contribuciones y se aumenta la calidad de servicio a los contribuyentes.



## **1.5. OBJETIVOS DE LA INVESTIGACIÓN**

### **1.5.1. Objetivo General**

Desarrollar un Sistema de Información con arquitectura Modelo Vista Controlador para la Recaudación tributaria por Obras Públicas de pistas y veredas de la Municipalidad Provincial de Puno.

### **1.5.2. Objetivos Específicos**

- Analizar y determinar los requerimientos del cliente para el dimensionamiento y la automatización del sistema.
- Desarrollar los módulos del Sistema de Información utilizando la técnica de desarrollo web AJAX, Framework CodeIgniter, metodología XP y los diagramas UML.
- Utilizar las librerías de seguridad del Framework CodeIgniter en la implementación del Sistema de Información.
- Evaluar el tiempo de atención a los contribuyentes.





## CAPÍTULO II

### REVISIÓN DE LITERATURA

#### 2.1. MARCO TEÓRICO

##### 2.1.1. Antecedentes De La Investigación

###### **Internacional**

Tapuy (2019), con el fin de disminuir costos y tiempo en las diferentes áreas de la empresa con la implementación del Sistema de Información con Arquitectura MVC para el Control de Inventario de Productos de la Empresa “Decorey Puyo”, concluye que es factible desarrollar los Sistemas de Información con Arquitectura MVC combinando la metodología XP y Sistema de producción Kanban, esto se demuestra en el desarrollo del proyecto del sistema de información de la empresa “Decorey Puyo”.

Gallardo (2018), en su trabajo de “Desarrollo de un sistema para Gestionar Ordenes de un Restaurante Usando MVC”, Concluye que el patrón de diseño MVC establece especial atención en cómo está organizada la aplicación, separándola en tres capas física y lógicamente.

De los Rios (2015), en su trabajo de Investigación de Modelo de adecuación y arquitectura de aplicaciones utilizando patrones arquitectónicos Modelo Vista Controlador en el frontend – Madrid España”, concluye que entre los principales beneficios se encuentran una reducción del 50% en el número de líneas de código, un gran incremento de la modularidad pasando a tener seis grandes módulos principales donde cada uno tiene una estructura modular, siguiendo el patrón Modelo Vista Controlador y un notable incremento de la robustez, y de la facilidad de mantenimiento y ampliación de funciones.



## Nacional

Valdivia (2018), en su investigación de **aplicación web basada en el patrón MVC para mejorar la gestión tributaria de la municipalidad distrital de Guadalupe, la libertad**, tiene como objetivo mejorar significativamente la gestión tributaria de la Municipalidad Distrital de Guadalupe con el desarrollo de una aplicación web, así como reducir el tiempo en la búsqueda de información de los contribuyentes, que finalmente concluye que la aplicación web, en la municipalidad Distrital de Guadalupe, influye en gran escala, puesto que permite reducir el tiempo en la búsqueda de información de los contribuyentes.

Tineo & Aguilar (2018), en su trabajo de investigación de **Propuesta de uso de Gobierno Electrónico para mejorar la Recaudación Tributaria en la Municipalidad Distrital de Pomalca, 2017**, se enfoca en el análisis y una propuesta de uso del gobierno electrónico para mejorar la recaudación tributaria de la municipalidad de Pomalca, teniendo como primer objetivo un análisis de la situación actual de la gerencia de administración tributaria y como segundo objetivo es determinar los factores influyentes que permiten una baja recaudación, llegando a la conclusión que los problemas de recaudación tributaria viene en un 40 % por la mala recaudación tributaria y que un **80%** por carecer de recursos tecnológicos y equipos informáticos.

Kawakami (2016), en su trabajo de **Propuesta de un sistema de recaudación tributaria del impuesto predial para optimizar la gestión pública de la Municipalidad provincial de Bagua Amazonas-2016**”, concluye que se debería implantar un Sistema de Recaudación Tributaria, esto permitiría optimizar mayores ingresos del Impuesto Predial, para obtener una adecuada Gestión y procesamiento de



datos, logrando ser más eficiente el servicio y también su recaudación que brinda en su respectiva área de Gestión Tributaria, así mismo se lograra una actualización del registro de datos de los contribuyentes pues por el momento existe muchas dificultades, tiene un sistema informático que no posee un control eficiente y eficaz, su estructura tributaria es antigua, le falta organización en cuanto al tema de recaudación del Impuesto Predial, existen deficiencias respecto al registro de los contribuyentes, sus sistemas no están actualizados pues esto genera que exista una inadecuada información, ya que no se da la información real y oportuna y esto no les permite una óptima toma de decisiones de sus funcionarios para cumplir con sus objetivos y metas.

Chuyes & Carreño (2014), **Desarrollo e Implementación del Sistema de Rentas De la Municipalidad Distrital de Bellavista Piura**”, concluye que se ha desarrollado un sistema que satisface completamente las necesidades del Usuario (Área de Rentas) dando así una herramienta óptima y benéfica para la Municipalidad Distrital de Bellavista para que pueda mejorar los procesos de recaudación Tributaria. Se mejoro notablemente los procesos que involucran la recaudación del impuesto predial, puesto que el sistema brinda información oportuna y confiable.

Con la implementación del nuevo sistema en la Unidad de Administración Tributaria se ha logrado una reducción promedio de 99% del tiempo consumido por los usuarios en el desarrollo de las actividades propias del Registro de Pagos (Predios).



## **Local**

Chucuya (2015), **Implementación de un sistema de información para la recaudación del impuesto de alcabala ‘SIPRIA’ de la Municipalidad Provincial de Puno -2015’**, concluye que la Implantación de un Sistema de Información (SIPRIA), permitió al personal realizar las actividades de manera eficiente la atención sea más óptimo en un 90% en la recaudación del Impuesto de Alcabala. Con la utilización de la Metodología Programación extrema, Lenguaje Unificado de Modelado, El Lenguaje de Programación CAKEPHP y Gestor de Base de Datos MySQL fueron herramientas fundamentales a la hora de desarrollar el software que sea mucho más rápida, legible, óptimo y confiable.

Barra & Limache (2017), en su trabajo de tesis **“Sistema de gestión para control de clientes morosos en banco "MI BANCO" del periodo 2015”**, concluye que desarrolló empleando las herramientas CodeIgniter, el cual es un framework PHP que emplea la arquitectura MVC (Modelo Vista Controlador) para la creación de aplicaciones ágiles, permitiendo desarrollar la aplicación con mayor rapidez, se empleó también un framework CSS, JavaScript, JQuery, Ajax entre otras herramientas que sirvieron para el proceso de implementación.

## **2.2. MARCO CONCEPTUAL**

### **2.2.1. Ley De Tributación Municipal**

En nuestro país tenemos del decreto legislativo N° 776, ley de Tributación Municipal, que precisamente en el capítulo III en los artículos 62 al 65, nos indica el marco normativo para la contribución especial de obras públicas.

### **Contribución Especial De Obras Públicas**



**Artículo 62°.** - La Contribución Especial de Obras Públicas grava los beneficios derivados de la ejecución de obras públicas por la Municipalidad. Las Municipalidades emitirán las normas procesales para la recaudación, fiscalización y administración de las contribuciones.

**Artículo 63°.** - En la determinación de la contribución especial por obras públicas, las Municipalidades calcularán el monto teniendo en consideración el mayor valor que adquiera la propiedad beneficiada por efecto de la obra municipal.

**Artículo 64°.** - En ningún caso las Municipalidades podrán establecer cobros por contribución especial por obras públicas cuyo destino sea ajeno a cubrir el costo de inversión total o un porcentaje de dicho costo, según lo determine el Concejo Municipal. Para efectos de la valorización de las obras y del costo de mantenimiento, las Municipalidades contemplarán en sus normas reglamentarias, mecanismos que garanticen la publicidad y la idoneidad técnica de los procedimientos de valorización, así como la participación de la población.

**Artículo 65°.** - El cobro por contribución especial por obras públicas procederá exclusivamente cuando la Municipalidad haya comunicado a los beneficiarios, previamente a la contratación y ejecución de la obra, el monto aproximado al que ascenderá la contribución.

### **2.2.2. Sistema de Información**

Un sistema es un conjunto de componentes que interrelacionan entre sí para lograr un objetivo común. Aunque existe una gran variedad de sistemas, la mayoría de ellos 20 pueden presentarse a través de un modelo formado por cinco bloques básicos; los siguientes elementos de entrada, elementos de salida. Sección de transformación, mecanismos de control y objetos. Los recursos acceden al sistema a través de los elementos de entrada para ser modificados en la sección de transformación. Este proceso



es controlado por el mecanismo de control con el fin de lograr el objetivo marcado. Una vez se ha llevado a cabo la transformación, el resultado sale del sistema a través de los elementos de salida. (Fernández, 2010)

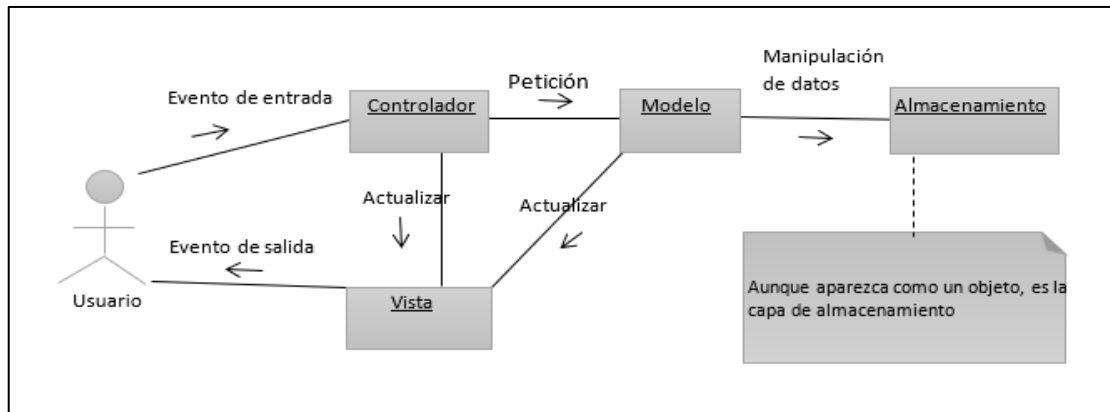
### 2.2.3. Modelo Vista Controlador (MVC)

El MVC es un patrón de diseño arquitectónico de *software*, que sirve para clasificar la información, la lógica del sistema y la interfaz que se le presenta al usuario. En este tipo de arquitectura existe un sistema central o controlador que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de buscar los datos e información necesaria y una interfaz que muestra los resultados al usuario final. Es muy usado en el desarrollo web porque al tener que interactuar varios lenguajes para crear un sitio es muy fácil generar confusión entre cada componente si estos no son separados de la forma adecuada. Este patrón permite modificar cada uno de sus componentes si necesidad de afectar a los demás.

**Modelo:** Este componente se encarga de manipular, gestionar y actualizar los datos. Si se utiliza una base de datos aquí es donde se realizan las consultas, búsquedas, filtros y actualizaciones.

**Vista:** Este componente se encarga de mostrarle al usuario final las pantallas, ventanas, páginas y formularios; el resultado de una solicitud. Desde la perspectiva del programador este componente es el que se encarga del frontend; la programación de la interfaz de usuario si se trata de una aplicación de escritorio, o bien, la visualización de las páginas web (CSS, HTML, HTML5 y JavaScript).

**Controlador:** Este componente se encarga de gestionar las instrucciones que se reciben, atenderlas y procesarlas. Por medio de el se comunican el modelo y la vista: solicitando los datos necesarios; manipulándolos para obtener los resultados; y entregándolos a la vista para que pueda mostrarlos. (García, 2017)



**Figura 1.** Modelo Vista Controlador (MVC)

Fuente: García, 2017

Con lo expuesto se resume las responsabilidades básicas del patrón MVC de la siguiente manera:

- El modelo es responsable de:
  - Acceder a la capa de almacenamiento de datos. Lo ideal es que el modelo independiente del sistema de almacenamiento.
  - Define las reglas del negocio.
- El controlador es responsable de:
  - Recibe los eventos de entrada.
  - En base a reglas de gestión de eventos se realizan peticiones al modelo o a las vistas.
- Las vistas son responsables de:
  - Recibir datos del modelo y los muestran al usuario.
  - Tienen un registro de su controlador asociado (normalmente porque además lo instancia).

Si una aplicación no está diseñada siguiendo el patrón de diseño MVC, inevitablemente se tendrá problemas de escalabilidad, mantenimiento y de extensión.



#### **2.2.4. Lenguaje Unificado de Modelado (UML)**

UML es un lenguaje de modelado, es decir, sirve para simplificar la realidad, para facilitar este modelado, se realiza una abstracción y se plasma en una gráfica, esto se conoce como modelado visual. UML sirve para el modelado completo de sistemas, tanto en el diseño de los sistemas software como para la arquitectura hardware donde se ejecuten. (Salvador, 2018)

#### **2.2.5. MariaDB**

**MariaDB** es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License). Por lo tanto, incorpora todas las mejoras de la versión de MySQL incorpora nuevos motores de almacenamiento mucho más eficientes que son Aria y XtraDB permitiendo ejecutar consultas más complejas y almacenarlas en cache y no en disco duro. (Auladell, 2016)

#### **2.2.6. HTML5**

HTML5 es la actualización de HTML, el lenguaje en el que es creada la web, HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de JavaScript. La versión anterior y más usada de HTML, carece de características necesarias para la creación de aplicaciones modernas basadas en un navegador. El uso fuerte de Java Script ha ayudado a mejorar esto, gracias a frameworks como jQuery, jQuery UI, entre otro. (Vega & Van Der Henst, 2011)

#### **2.2.7. PHP**

Según php.net, PHP es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML5. PHP es un lenguaje de script que se ejecuta del lado del servidor, el código incluye en una página HTML normal. Por lo tanto, se puede comparar con otros lenguajes de script





que se ejecutan según el mismo principio: ASP (Active Server Pages), JSP (Java Server Pages), a diferencia de un lenguaje como JavaScript, donde el código se ejecuta del lado del cliente (en el explorador), el código PHP se ejecuta del lado del servidor. El resultado de esta ejecución se incrusta en la página HTML, que se envía al navegador. (Heurtel , 2016).

### **2.2.8. JavaScript**

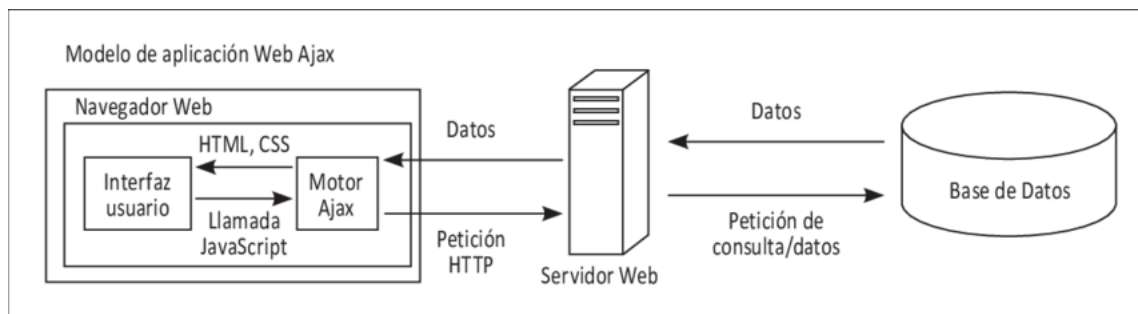
JavaScript es un lenguaje de programación, comparable con cualquier otro lenguaje de programación profesional como C++ o Java. JavaScript difiere de los demás lenguajes en que puede realizar tareas personalizadas, desde almacenar valores hasta calcular algoritmos complejos, incluida la capacidad de interactuar con los elementos del documento y procesar su contenido dinámicamente. Al igual que HTML y CSS, JavaScript se incluye en los navegadores y, por lo tanto, se encuentra disponible en todos nuestros documentos. (Gauchat, 2017)

### **2.2.9. AJAX**

Ajax es el acrónimo de Asynchronous JavaScript and XML, el termino fue invitado por Jesse James Garrett y apareció por primera vez en su ensayo de febrero de 2005: “Ajax: A New Approach to Web Application”, El termino fue inventado por el Jesse, pero el objeto XMLHttpRequest fue introducido en el mercado por Microsoft a mediados de 1999.

Ajax es un conjunto de tecnología: JavaScript, XML y un lenguaje del lado del servidor que no está contenida en el nombre pero que es igualmente importante. Al comienzo, los datos se enviaban principalmente en el formato XML, pero con la creación y popularidad del formato JSON este último paso a ser el formato dominante. La idea esencial del Ajax

es hacer una petición al servidor sin tener que renderizar toda la página HTML, lo que era costoso en su época debido a que la mayoría de las conexiones con internet eran bastante lenta. Pero lo mejor es que esto está integrado con el lenguaje JavaScript. Esta combinación explosiva dio origen a una nueva era de aplicaciones web. (Ayoze, 2017)



**Figura 2.** Modelo de Aplicación Web Ajax

Fuente: researchgate.net

### 2.2.10. NGINX

NGINX es un servidor HTTP gratuito, de código abierto, de alto rendimiento y proxy inverso, así como un servidor proxy IMAP / POP3. NGINX es conocido por su alto rendimiento, estabilidad, amplio conjunto de funciones, configuración simple y bajo consumo de recursos.

NGINX es uno de los pocos servidores escritos para abordar el problema C10K. A diferencia de los servidores tradicionales, NGINX no depende de hilos para manejar las solicitudes. En su lugar, utiliza una arquitectura mucho más escalable basada en eventos (asíncrona). Esta arquitectura utiliza pequeñas, pero más importante, cantidades predecibles de memoria bajo carga. Incluso si no espera manejar miles de solicitudes simultáneas, aún puede beneficiarse del alto rendimiento y la pequeña huella de memoria de NGINX. NGINX escala en todas las direcciones: desde el VPS más pequeño hasta los grandes grupos de servidores. (Nginx, 2017)



### 2.2.11. CodeIgniter

CodeIgniter es un Framework, un marco de desarrollo de aplicaciones, un conjunto de herramientas para crear sitios web utilizando PHP. Es un producto de código libre, libre de uso para cualquier aplicación. Su objetivo es permitir desarrollar proyectos mucho más rápido, proporcionando un amplio conjunto de bibliotecas para las tareas más comunes, así como una interfaz siempre y una estructura lógica para acceder a estas bibliotecas. (CodeIgniter, 2018)

Este framework permite desarrollar proyectos mucho más rápido de lo que se podría hacer si se los escribiese desde cero, para esto provee muchas librerías para tareas repetitivas y necesarias, así como una interface simple y una estructura lógica para acceder a esas librerías. CodeIgniter permite enfocarse principalmente en el proyecto minimizando la cantidad de código necesario para una tarea dada.

CodeIgniter se encuentra bajo una licencia open source y es compatible con PHP5 y PHP7 y está basado en el patrón de diseño Modelo -Vista -Controlador que permite separar el diseño con el código, agilizar las funciones a la base de datos y crear librerías independientes para ser reutilizadas en cualquier instancia de la aplicación. Al utilizar librerías en una aplicación se logra agilizar y optimizar la aplicación a nivel web.

CodeIgniter viene con un conjunto de librerías que permiten realizar las tareas de desarrollo web típicamente necesarias, como: acceder a una base de datos, enviar un e-mail, validar datos de un formulario, mantener sesiones, manipular imágenes, trabajar con datos XML-JSON, etc. El sistema puede ser fácilmente extendido a través de la inclusión de plugins y librerías, o a través de extensión de clases o ganchos del sistema.



## CAPÍTULO III

### MATERIALES Y MÉTODOS

#### 3.1. POBLACIÓN Y MUESTRA

##### POBLACIÓN

La población de usuarios para el Sistema de Información estuvo conformada por **5** trabajadores, que están directamente relacionados con todo el proceso de recaudación tributaria por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno.

La población de contribuyentes fue (**680**) personas que son beneficiarios de obras públicas específicamente de pistas y veredas.

##### MUESTRA

La muestra de contribuyentes por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno fueron **66** contribuyentes. Para precisar el tamaño de muestra se usó el método de muestreo por afijación proporcional. Donde la unidad de muestreo son los contribuyentes que se apersonaron a pagar su deuda.

##### Cálculo Tamaño De Muestra

Dado que la población es finita, pues sabemos el total de la población y deseamos estar al tanto de cuantos tendremos que estudiar, se formuló de esta manera:

En el que:

- $N$  = Total de la población.
- $Z$  = Valor tabular.
- $p$  = Probabilidad de que ocurra el evento estudiado  $5\% = 0.05$ .
- $q = 1 - p$  probabilidad de que no ocurra el evento estudiado.
- $E$  = Error de muestra  $5\%$ .

$$n = \frac{N \times Z_{\alpha}^2 \times p \times q}{E^2 \times (N - 1) + Z_{\alpha}^2 \times p \times q} \quad (1)$$

$$n = \frac{680 \times 1.96^2 \times 0.95 \times 0.05}{0.05^2 \times (680 - 1) + 1.96^2 \times 0.95 \times 0.05}$$

$$n = \frac{124.08368}{1.879976}$$

$$n = 66.0028$$

$$n = 66$$

Por lo tanto, se obtuvo el tamaño de muestra  $n = 66$  contribuyentes para la investigación. La obtención de la unidad de muestreo se realizó en función no aleatoriamente a los contribuyentes que se apersonaron a La Subgerencia De Promoción Social Y Participación Ciudadana a realizar el pago.

### 3.2. TIPO DE INVESTIGACIÓN

Hernández Sampieri et al. (2017), indica que una investigación es de tipo no experimental cuando no se manipulan las variables, por lo contrario, se toma la información tal como se muestra en la realidad.

Ayudó en el desarrollo de sistema de información, en la recolección de datos donde el propósito fue describir las variables y analizar su incidencia e interrelación en un momento dado.

### 3.3. DISEÑO DE LA INVESTIGACIÓN

#### PRE EXPERIMENTAL

Según Hernández Sampieri et al. (2017), se llama así porque su grado de control es mínimo. Diseño de pre-prueba/post-prueba con un solo grupo: a un grupo se aplica una prueba previa al estímulo o tratamiento experimental, después se le administra el tratamiento y finalmente se le aplica una prueba posterior al estímulo.



**Figura 3.** Pre Prueba y Post Prueba

Fuente: Valdivia, 2018

Donde:

G: Grupo Experimental

O1: Gestión Tributaria en la Municipalidad Provincial de Puno **anterior** a la implementación del sistema de información basada en el patrón MVC.

X: Sistema de información

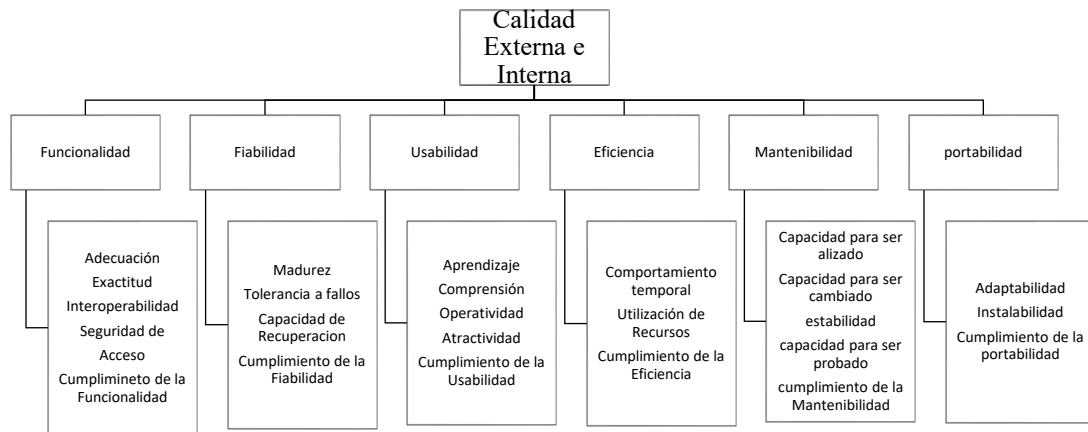
O2: Gestión Tributaria en la Municipalidad Provincial de Puno **posterior** de la implementación del sistema de información basada en el patrón MVC.

### 3.4. MÉTODO DE VALIDACIÓN DE SOFTWARE

El mundo globalizado exige cada vez más la aplicación de estándares internacionales que garanticen la calidad de los productos. Por esta razón, es necesario

que todo aquel que se dedica al desarrollo de software incluya en sus procesos, estándares de calidad que permitan certificarse en alguno de los modelos. (Abud, 2012)

El modelo de calidad ISO-9126, clasifica la calidad del Software en un conjunto estructurado de características y sub-características.



**Figura 4.** Modelo de Calidad del Software ISO – 9126

Fuente: Estándar ISO – 9126

Para la validación de la calidad del producto de software del sistema “SIRPACEPV”, se empleó la ficha de evaluación del estándar ISO-9126, en el cual se recogió las respuestas emitidas por los usuarios del sistema.

### 3.5. MÉTODO DE RECOPIACIÓN DE DATOS

Para realizar el trabajo de investigación se aplicó una encuesta (ver **anexo A**) a los contribuyentes que si con la implementación de un sistema de información se mejoraría la gestión administrativa de recaudación tributaria. Otra encuesta a los trabajadores sobre el funcionamiento del sistema de información (ver **anexo B**). Además, se realizó otra encuesta a los contribuyentes sobre el tiempo que demora para realizar sus pagos antes y después de implementar el sistema en la oficina de Sub Gerencia de

Promoción Social y Participación Ciudadana - Municipalidad Provincial de Puno (ver **anexo C**),

Es el concepto de interrogaciones ordenadas, encaminadas a un modelo específico de una población con la finalidad de saber cambios de opinión o hechos determinados, por tanto, se aplicó en esta investigación, considerando que la encuesta está dirigida a los trabajadores y contribuyentes.

Para la evaluación de la calidad del producto de software, las respuestas fueron recopiladas en la ficha de evaluación de la calidad del producto de software estándar ISO-9126.

### 3.6. MÉTODO DE TRATAMIENTO DE DATOS

#### Cálculo De Diferencia Entre Dos Medias Con Observaciones Apareadas

Sean  $(X_1; Y_1), (X_2; Y_2), \dots, (X_n; Y_n)$ , una muestra aleatoria de  $n$  datos apareados, donde las muestras  $X_1, \dots, X_n$ , e  $Y_1, \dots, Y_n$ , correlacionadas, son seleccionadas respectivamente de dos poblaciones normales  $X \sim N(\mu_1, S_1^2)$  y  $Y \sim N(\mu_2, S_2^2)$ .

Podemos concebir esta  $n$  diferencias:

$$D_1 = X_1 - Y_1, D_2 = X_2 - Y_2, \dots, D_n = X_n - Y_n \quad (2)$$

Como una muestra aleatoria seleccionada de una población de diferencias  $D = X - Y$  cuya distribución normal  $N(\mu_D, S_D^2)$ , con media  $\mu_D = \mu_1 - \mu_2$  y varianza  $S_D^2 = S_1^2 + S_2^2 - 2cov(X, Y)$ .

Si  $S_D^2$  es conocida, la estadística  $\bar{D}$ , media de las diferencias tiene distribución normal  $N(\mu_D, S_D^2/n)$ . Consecuentemente la estadística  $Z \sim N(0; 1)$ , esta estadística  $Z$  se utiliza en





la prueba de dos medias correlacionadas cuando la varianza  $S_D^2$  es conocida. (Cordova, 2003)

La base de pruebas para la comparación de medias apareadas consiste en analizar las diferencias entre las observaciones de un mismo individuo.

Suponiendo que la variable aleatoria que define la diferencia entre dos observaciones registradas en un mismo individuo (modelo antes-después) fuera una variable aleatoria que se distribuyera normalmente, y queremos contrastar la hipótesis de que se produjo un efecto entre ambas observaciones.

En base a las pruebas de comparación de medias apareadas, se realizó un contraste a la hipótesis de que se produjo un efecto entre ambas observaciones (Antes-Después).

#### **i. Planteamiento De Hipótesis**

$H_0: \mu_x \geq \mu_y$  (Con la implementación del Sistema de Información con Arquitectura MVC, para la recaudación tributaria por obras públicas de pistas y veredas para la oficina de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno, no se reduce el tiempo de atención a los contribuyentes).

$H_0: \mu_x < \mu_y$  (Con la implementación del Sistema de Información con Arquitectura MVC, para la recaudación tributaria por obras públicas de pistas y veredas para la oficina de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno, se reduce el tiempo de atención a los contribuyentes).

## ii. Nivel De Significancia

Se uso el Nivel de significancia  $\alpha = 0.05 = 5\%$  y un valor tabular  $z_t$ , en este caso se tiene  $z_t = 1.645$ .

## iii. Prueba Estadística

$$Z_c = \frac{\bar{d}}{S_d/\sqrt{n}} \sim N(0; 1) \quad (3)$$

Donde:

$$\bar{d} = \frac{\sum_{i=1}^n d_i}{n} = \bar{x}_{post} - \bar{y}_{pre} \quad d_i = x_i - y_i, i = 1, 2, \dots, n$$
$$S_d^2 = \frac{\sum_{i=1}^n (d_i - \bar{d})^2}{n - 1} \quad (4)$$

$Z_c$ : Z calculada.

$\bar{d}$ : Promedio muestral de la diferencia de resultados (Antes y Después).

$n$ : Tamaño de muestra del grupo en estudio.

$S_d$ : Desviación estándar de la diferencia de (Antes y Después).

$y$ : Tiempo de atención antes de la implementación del sistema "SIRPACEPV".

$x$ : Tiempo de atención después de la implementación del sistema "SIRPACEPV".

## iv. Regla De Decisión

Si  $Z_c > Z_t$ , entonces se rechaza la  $H_0$  y se acepta la  $H_a$

## v. Conclusión

Dependiendo del resultado de la regla de decisión, se dará una interpretación acerca de los datos analizados.



### 3.7. MATERIAL EXPERIMENTAL

El presente trabajo se ejecutó tomando en consideración la información de la Oficina encargada de la administración por contribución especial de obras públicas de pistas y veredas, los materiales principales son:

**Padrón De Contribuyentes:** Donde se encuentran todos los datos personales necesarios del contribuyente.

**Base De Datos:** Donde se encuentran los datos de los contribuyentes ya registrados anteriormente.

**Documentos en Word:** Donde se encuentran todas las plantillas de documentos que se entrega al contribuyente por la oficina.

#### 3.7.1. Hardware

Se utilizó una computadora portátil con procesador Intel Core i7, RAM 12GB, Disco Solido 450 GB.

#### 3.7.2. Software

Para el funcionamiento y desarrollo del sistema de información se realizó con estas herramientas:

##### Herramientas de Desarrollo

- Visual Studio - editor de código fuente.
- PHP7 – Lenguaje de programación.
- JavaScript – Lenguaje de programación.
- MariaDB – Sistema de gestión de bases de datos.
- CodeIgniter – Framework PHP.
- Google Chrome - Navegador web.
- StarUML the open source UML/MDA.



## Herramientas de Ejecución

- Debian 10 – Servidor Linux.
- Nginx - Servidor web.
- PHP7 – Lenguaje de Programación.
- MariaDB – Sistema de gestión de bases de datos.
- Google Chrome – Navegador web

### 3.7.3. Material De Escritorio

- ✓ Documentos y datos referenciales acerca del proceso de cobro por contribución especial por obras públicas de pistas y veredas.
- ✓ Reglamentos, de organización y funciones.
- ✓ Ley de tributación Municipal.

## 3.8. METODOLOGÍA DE DESARROLLO DEL SISTEMA

### 3.8.1 Metodología de desarrollo de software (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo, corto equipo, la metodología se basa en el usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Los objetivos de XP en realidad profundizan en la satisfacción del cliente, proporcionando el software que necesita y cuando lo requiera.

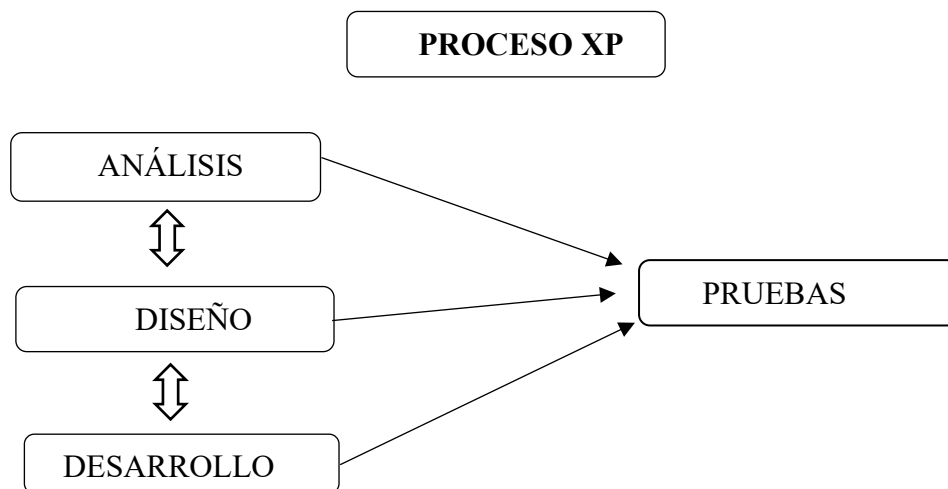
#### Valores XP

- Comunicación
- Simplicidad
- Retroalimentación
- Valentía

- Respeto

En el desarrollo del sistema de información se sigue obligatoriamente el proceso XP, mediante ciclos de desarrollo cortos (llamado iteraciones) donde se generarán los entregables funcionales.

En cada iteración se realiza un ciclo completo de análisis, diseño, desarrollo y pruebas, conjuntamente con el lenguaje Unificado de Modelado (UML) y la Arquitectura Cliente / Servidor.



**Figura 5.** Proceso XP

Fuente: <https://www.diegocalvo.es>

**Análisis:** El análisis comienza escuchando actividad para recabar requerimientos que permite que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y adquieran la sensibilidad de la salida y características principales y funcionalidad que se requieren.

Escuchar lleva a la creación de historias por parte del usuario, estas son tomadas por los desarrolladores para modelar los requisitos. (Pressman, 2010)

**Diseño:** Crea un “bosquejo” del objeto por hacer a fin de entender el panorama general. Un ingeniero de software crea modelos a fin de entender mejor los requerimientos del software y el diseño que los satisfará. (Pressman, 2010)



**Desarrollo:** Esta actividad combina la generación de código (ya sea manual o automatizada) para generar los entregables funcionales. (Cortés, 2005)

**Pruebas:** Se utiliza para descubrir errores. Los desarrolladores escriben pruebas unitarias y los clientes especifican las pruebas funcionales.



## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

El sistema de información para la oficina de Promoción Social y Participación ciudadana de la Municipalidad Provincial de Puno, se desarrolló e implementó principalmente para mejorar la gestión administrativa de recaudación tributaria por Obras Públicas de pistas y veredas, mediante la automatización de actividades como el registro de contribuyentes y los pagos que realiza, reportes, documentos de constancias de no adeudo de manera oportuna. El manejo de información de manera manual y el aumento de contribuyentes cada año por obra ejecutada, ha originado al personal administrativo mayor carga laboral y problemas de pérdida de información originando errores de cálculo, lo que ha originado a recurrir a nuevas tecnologías para agilizar la gestión administrativa principalmente la atención a los contribuyentes.

Todo lo expuesto anteriormente fueron evidencias para el desarrollo e implementación de un sistema de información para brindar una atención más eficiente a los contribuyentes.

#### 4.1 ANÁLISIS DEL SISTEMA

Para desarrollar el sistema “SIRPACEPV”, se realizó procedimientos que permitió realizar de forma ordenada las etapas de elaboración y ejecución de acuerdo a la metodología de desarrollo de software XP y el lenguaje Unificado de Modelado (UML).

##### 4.1.1 Análisis de Requerimientos

A continuación, se tiene las historias de usuario que son producto del análisis de requisitos para el desarrollo del software:



### Tabla 1.

#### *Historia de Usuario - Autenticación de Usuario*

---

<b>Historia de Usuario</b>	
<b>Número: 1</b>	<b>Usuario: Operador del Sistema</b>
<b>Nombre historia:</b> Autenticación de Usuario	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> (Alta / Media / Baja)
<b>Puntos estimados: 3</b>	<b>Iteración asignada: 1</b>
<b>Programador responsable:</b> Juan Carlos Huanca Mollinedo	

---

*Fuente: Elaboración Propia*

#### **Descripción:**

Los usuarios deben identificarse para acceder a cualquier parte del sistema. La autenticación permitirá saber si es realmente el encargado de gestionar el sistema y una vez validado se debe determinar el nivel de acceso a diferentes módulos. Se debe registrar la siguiente información:

- Apellidos y nombres del operador del sistema.
- DNI.
- Número de contacto.
- Correo electrónico.
- Tipo de usuario.
- Estado (Activo / Desactivo).
- Usuario.
- Contraseña.





**Tabla 2.**

*Historia de Usuario - Registrar Obras*

---

**Historia de Usuario**

---

<b>Número: 2</b>	<b>Usuario: Operador del Sistema</b>
------------------	--------------------------------------

---

**Nombre historia:** Registrar Obras

---

<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> (Alta / Media / Baja)
--------------------------------------	-------------------------------------------------------

---

<b>Puntos estimados: 3</b>	<b>Iteración asignada: 2</b>
----------------------------	------------------------------

---

**Programador responsable:** Juan Carlos Huanca Mollinedo

---

*Fuente: Elaboración Propia*

**Descripción:**

El sistema debe ser capaz de permitir al(usuario) ingresar la información de la obra en ejecución con los siguientes atributos:

- Nombre de la obra.
- Año y mes de ejecución.
- Observaciones de la obra.

El sistema debe permitir al usuario hacer búsqueda en tiempo real y filtros con palabras claves para identificar a cada obra.

**Acciones requeridas:** Visualizar, modificar, eliminar.



**Tabla 3.**

*Historia de Usuario - Registrar Contribuyente*

---

<b>Historia de Usuario</b>	
<b>Número: 3</b>	<b>Usuario: Operador del Sistema</b>
<b>Nombre historia:</b> Registrar Contribuyente	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> (Alta / Media / Baja)
<b>Puntos estimados: 3</b>	<b>Iteración asignada: 2</b>
<b>Programador responsable:</b> Juan Carlos Huanca Mollinedo	

---

*Fuente: Elaboración Propia*

**Descripción:**

El sistema debe registrar al contribuyente con toda la información requerida.

**Atributos:**

- Apellido y Nombres.
- Dirección de la vivienda.
- Tipo de vía (jr., pasaje, avenida).
- Longitud de frentera de la vivienda.
- Ancho de vía en ml.
- Ancho de la vereda en ml.
- Observaciones acerca del contribuyente.

El sistema debe permitir al usuario hacer búsqueda en tiempo real y filtros con palabras claves para identificar a los contribuyentes.

**Acciones requeridas:** Visualizar, modificar, eliminar.



**Tabla 4.**

*Historia de Usuario - Generar Monto de Contribución*

---

**Historia de Usuario**

---

**Número: 4**

**Usuario: Operador del Sistema**

---

**Nombre historia:** Generar Monto de Contribución

---

**Prioridad en negocio:**  
Alta

**Riesgo en desarrollo:**  
(Alta / Media / Baja)

---

**Puntos estimados: 3**

**Iteración asignada: 3**

---

**Programador responsable:** Juan Carlos Huanca Mollinedo

---

*Fuente: Elaboración Propia*

---

**Descripción:**

El sistema debe calcular el monto total de contribución a partir de la información registrada del contribuyente y de la obra.

Finalmente, una vez calculada el monto total, el sistema debe guardar esa información en la base de datos e imprimir un documento con información necesaria para el contribuyente.

**Acciones requeridas:**

Filtrar información de propietarios por apellidos y nombres.



**Tabla 5.**

*Historia de Usuario - Registrar pagos*

---

**Historia de Usuario**

---

**Número: 5**

**Usuario: Operador del Sistema**

---

**Nombre historia:** Registrar pagos

---

**Prioridad en negocio:**  
Alta

**Riesgo en desarrollo:**  
(Alta / Media / Baja)

---

**Puntos estimados: 3**

**Iteración asignada: 4**

---

**Programador responsable:** Juan Carlos Huanca Mollinedo

---

*Fuente: Elaboración Propia*

---

**Descripción:**

El sistema debe generar un recibo con los datos del contribuyente y el monto en soles que pagará en caja.

Después de que el contribuyente realice su pago en caja de la Municipalidad Provincial de Puno, el sistema debe de registrar el monto del pago que realizó y restar al monto total generado del contribuyente y finalmente almacenar el comprobante de pago en un archivo PDF O JPG.

**Acciones requeridas:**

Filtrar información de propietarios por apellidos y nombres.



**Tabla 6.**

*Historia de Usuario - Generar Constancias de cancelación y no adeudo.*

---

**Historia de Usuario**

---

**Número: 6**

**Usuario: Operador del Sistema**

---

**Nombre historia:** Generar Constancias de cancelación y no adeudo.

---

**Prioridad en negocio:**

Alta

**Riesgo en desarrollo:**

(Alta / Media / Baja)

---

**Puntos estimados: 3**

**Iteración asignada: 5**

---

**Programador responsable:** Juan Carlos Huanca Mollinedo

---

*Fuente: Elaboración Propia*

---

**Descripción:**

El sistema debe generar los siguientes documentos:

- **Constancia de cancelación** detallando todos los pagos que realizo en contribuyente.
- **Constancia de no adeudo** indicando que el contribuyente ya no tiene deudas y completó su pago.

**Acciones requeridas:**

Filtrar información de propietarios por apellidos y nombres.



**Tabla 7.**

*Historia de Usuario - Generar Notificaciones.*

---

<b>Historia de Usuario</b>	
<b>Número: 7</b>	<b>Usuario: Operador del Sistema</b>
<b>Nombre historia:</b> Generar Notificaciones.	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> (Alta / Media / Baja)
<b>Puntos estimados: 3</b>	<b>Iteración asignada: 6</b>
<b>Programador responsable:</b> Juan Carlos Huanca Mollinedo	

---

*Fuente: Elaboración Propia*

**Descripción:**

Para los contribuyentes que mantienen deuda el sistema debe ser capaz de generar notificación en un documento PDF con los datos del contribuyente y con la deuda que aún mantiene.

**Acciones requeridas:**

Filtrar información de obras.



**Tabla 8.**

*Historia de Usuario – Visualizar archivos generados*

---

<b>Historia de Usuario</b>	
<b>Número: 8</b>	<b>Usuario: Operador del Sistema</b>
<b>Nombre historia:</b> Visualizar archivos generados	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> (Alta / Media / Baja)
<b>Puntos estimados: 3</b>	<b>Iteración asignada: 7</b>
<b>Programador responsable:</b> Juan Carlos Huanca Mollinedo	

---

*Fuente: Elaboración Propia*

**Descripción:**

El sistema de información debe tener un módulo para visualizar todos los documentos generados para cada contribuyente en formato digital.

**Acciones requeridas:**

Filtrar información de propietarios por apellidos y nombres.



**Tabla 9.**

*Historia de Usuario - Ver y generar Reporte de pagos*

---

<b>Historia de Usuario</b>	
<b>Número: 9</b>	<b>Usuario: Operador del Sistema</b>
<b>Nombre historia:</b> Ver y generar Reporte de pagos	
<b>Prioridad en negocio:</b> Alta	<b>Riesgo en desarrollo:</b> (Alta / Media / Baja)
<b>Puntos estimados: 3</b>	<b>Iteración asignada: 8</b>
<b>Programador responsable:</b> Juan Carlos Huanca Mollinedo	

---

*Fuente: Elaboración Propia*

**Descripción:**

El sistema debe generar reportes de cada obra con datos de los contribuyentes y la suma total de pagos realizados, el reporte debe ser de tipo general y por rango de fechas de la misma manera deben ser filtrados por contribuyentes (sin deuda, con deuda) y finalmente imprimir en un archivo PDF el reporte generado.

**4.1.2. Iteraciones**

Se han definido iteraciones para cubrir las necesidades planteadas, en cada una de ellas se ha incluido las historias de usuario en función de su prioridad de construcción, esto se detalla en la tabla:



**Tabla 10.**

*Tabla de ilustraciones.*

<b>Iteración</b>	<b>Historia de usuario</b>
1	- Autenticación de usuario.
2	- Registrar obras. - Registrar contribuyente.
3	- Generar monto de contribución.
4	- Registrar pagos.
5	- Generar constancias de cancelación y no adeudo.
6	- Generar notificaciones.
7	- Visualizar archivos generados.
8	- Visualizar y generar reporte de pagos.

*Fuente: Elaboración Propia.*

#### **4.1.3 Requerimientos no funcionales**

Se presenta los requerimientos no funcionales que debe cumplir el sistema de información:

**Usabilidad:** El sistema debe presentar una interfaz de usuario de uso intuitiva y sencilla.

**Disponibilidad:** El sistema tendrá que estar en funcionamiento las 24 horas y los 7 días de la semana. Ya que el sistema está diseñado para la carga de datos y constante comunicación con los contribuyentes.

**Escalabilidad:** El sistema debe tener la capacidad de soportar el aumento de datos y procesarlas con ningún problema sin afectar significativamente su velocidad de respuesta.

**Mantenibilidad:** El sistema debe ser capaz de atender nuevos requerimientos que se presenten en el futuro adaptarse al mundo cambiante de hoy.

**Seguridad:** El sistema debe ser desarrollada aplicando patrones y recomendaciones de programación que incrementen a la seguridad de datos.

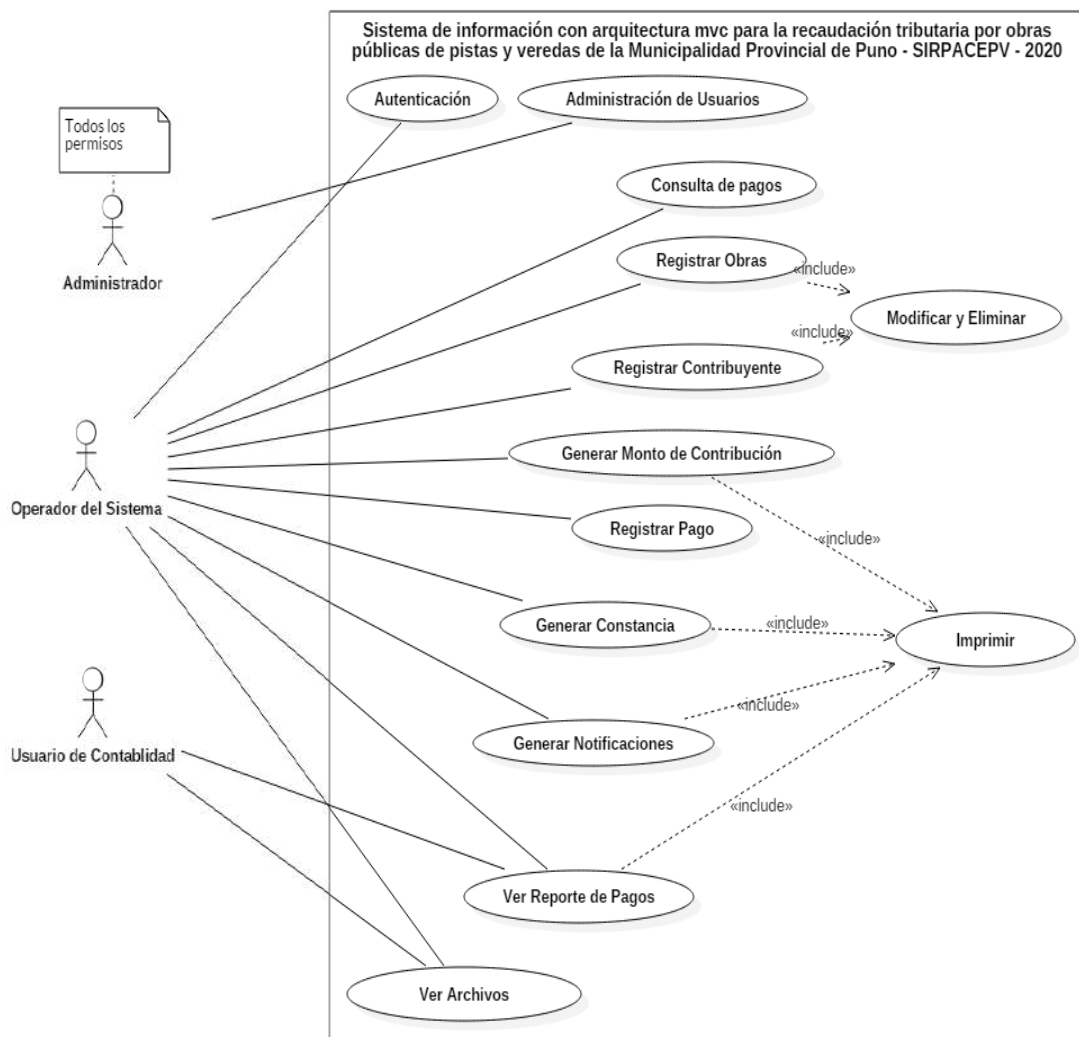
Los permisos de acceso al servidor o a la base de datos solo deben ser autorizados por el ente máximo de la oficina quien tendrá todas las claves de acceso de tipo administrador.

## 4.2 DISEÑO DEL SISTEMA

Para entender el modelamiento en el diseño del sistema se utilizó el lenguaje Unificado de Modelado (UML), la misma que ha permitido obtener los siguientes diagramas.

### 4.2.1 Diagrama de Casos De Uso

Mediante este diagrama se representa al sistema en su forma general y los privilegios que posee los usuarios del sistema.



**Figura 6.** Diagrama de Casos De Uso

Fuente: Elaboración Propia.



**Autenticación:** Permite identificar al usuario encargado de manipular el sistema.

**Consulta de Pagos:** Le permite realizar consultas de pagos de contribuyentes de manera detallada.

**Registrar Obra:** Tiene la capacidad de registrar nuevas obras de pavimentación de pistas y veredas.

**Registrar Contribuyente:** Registra al sistema nuevos contribuyentes.

**Generar Monto de Contribución:** Realiza el cálculo del monto de contribución según la información de medidas del inmueble del contribuyente.

**Registrar Pagos:** Le permite registrar los pagos de contribuyentes de manera detallada.

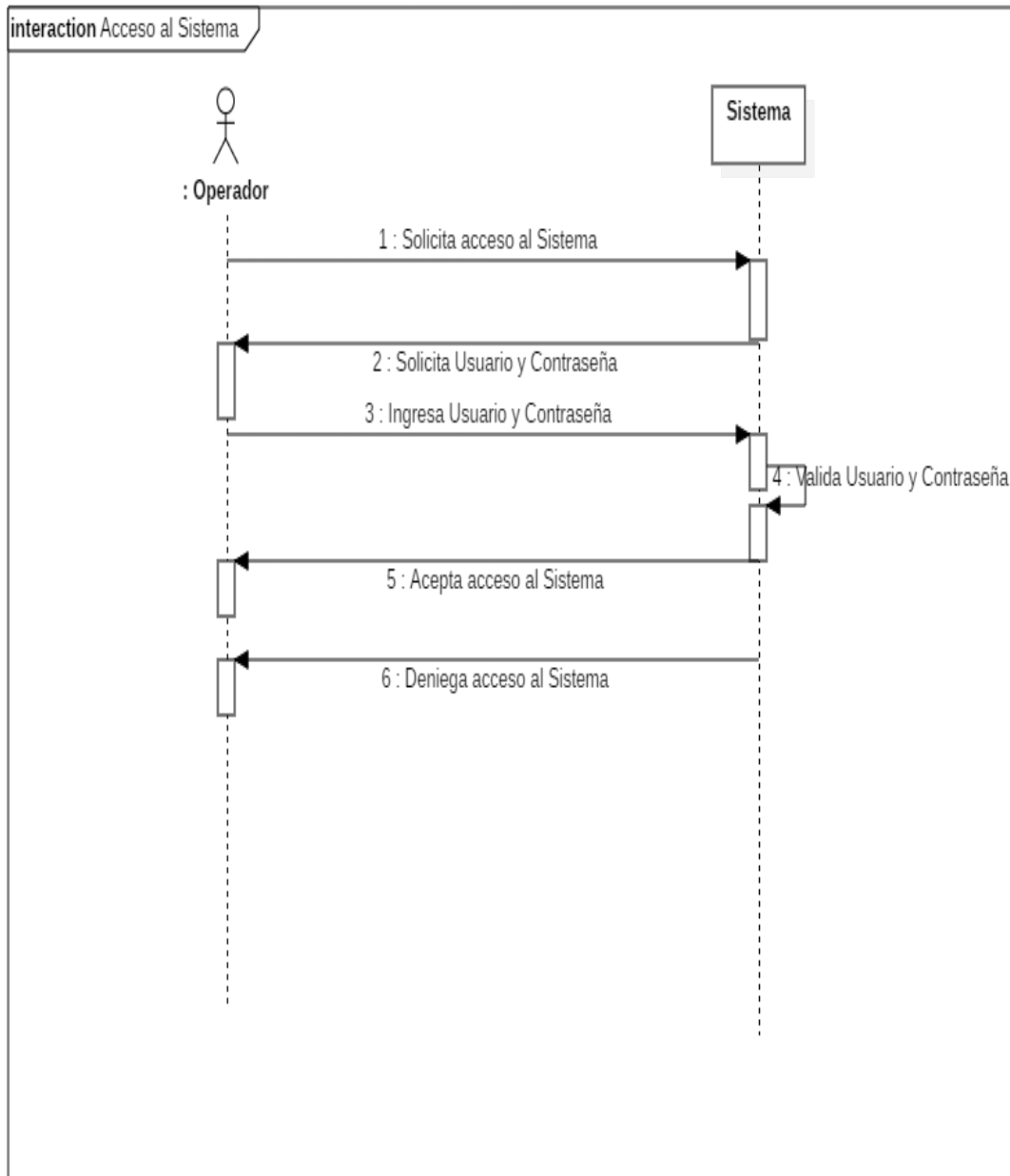
**Generar Constancias:** Genera Constancias de no adeudo y constancias de cancelación siempre en cuando el contribuyente cumpla con el total de su deuda.

**Generar Notificaciones:** Genera notificaciones a los contribuyentes que mantienen una deuda.

**Ver Archivos:** Visualiza los archivos generados de cada contribuyente documentos, recibos de pago, constancias etc.

#### 4.2.2 Diagrama de Secuencia

**Interfaz para el acceso al sistema:** Para ingresar al sistema el operador solicita su acceso, en respuesta el sistema solicita que ingrese usuario y contraseña para validar los datos ingresados para aceptar o denegar su acceso.



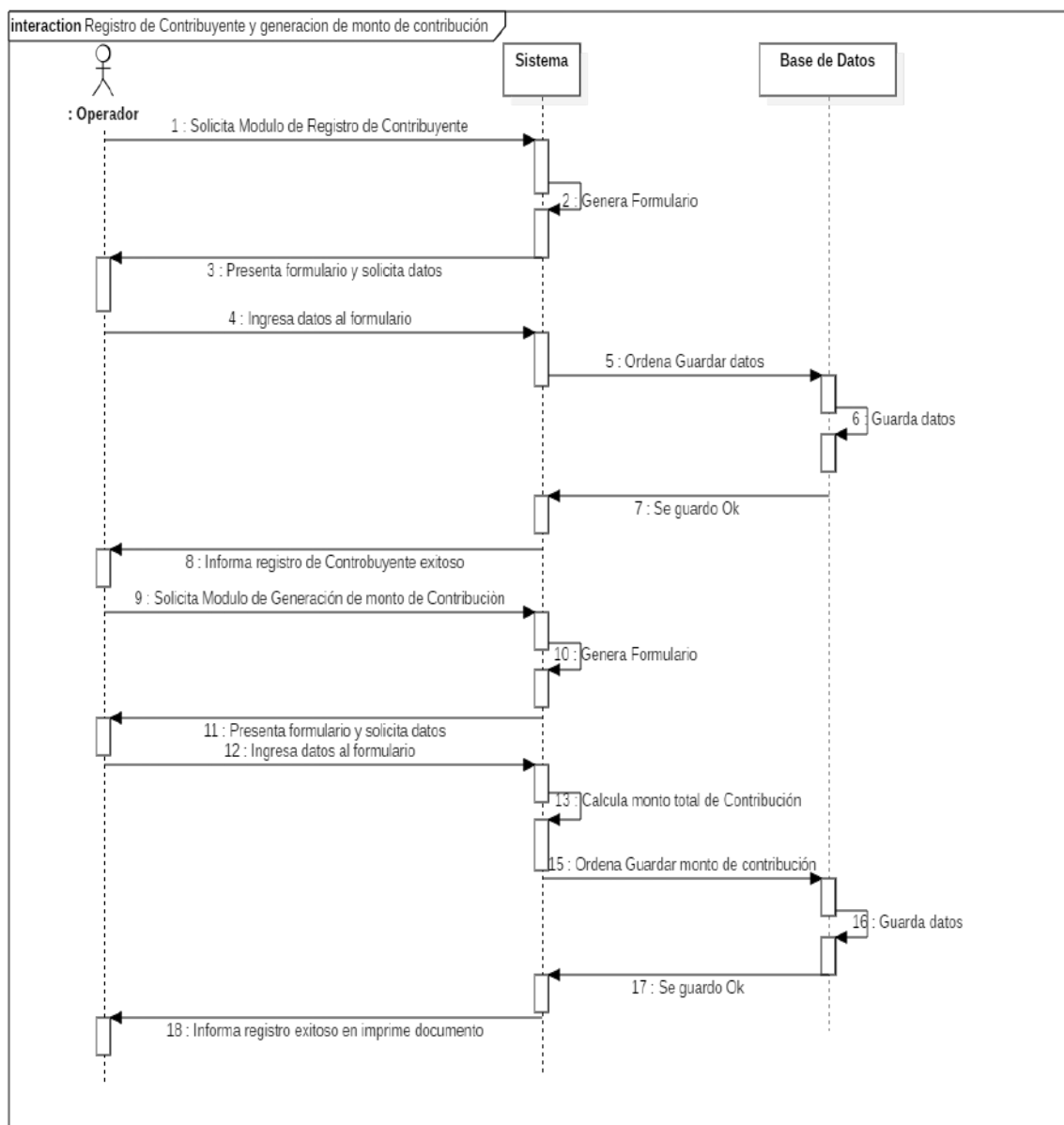
**Figura 7.** Diagrama de Secuencia - Acceso al Sistema

Fuente: Elaboración Propia

**Interfaz para registrar contribuyente y generar monto de contribución:** Se muestra la secuencia de interacciones realizadas para poder registrar un nuevo contribuyente y generar el monto total de contribución para ello se hace uso de información recogida del contribuyente. El sistema muestra un formulario para llenar datos del contribuyente, una

vez completado el formulario con los datos correctos el sistema verifica si son correctos y almacena en la base de datos.

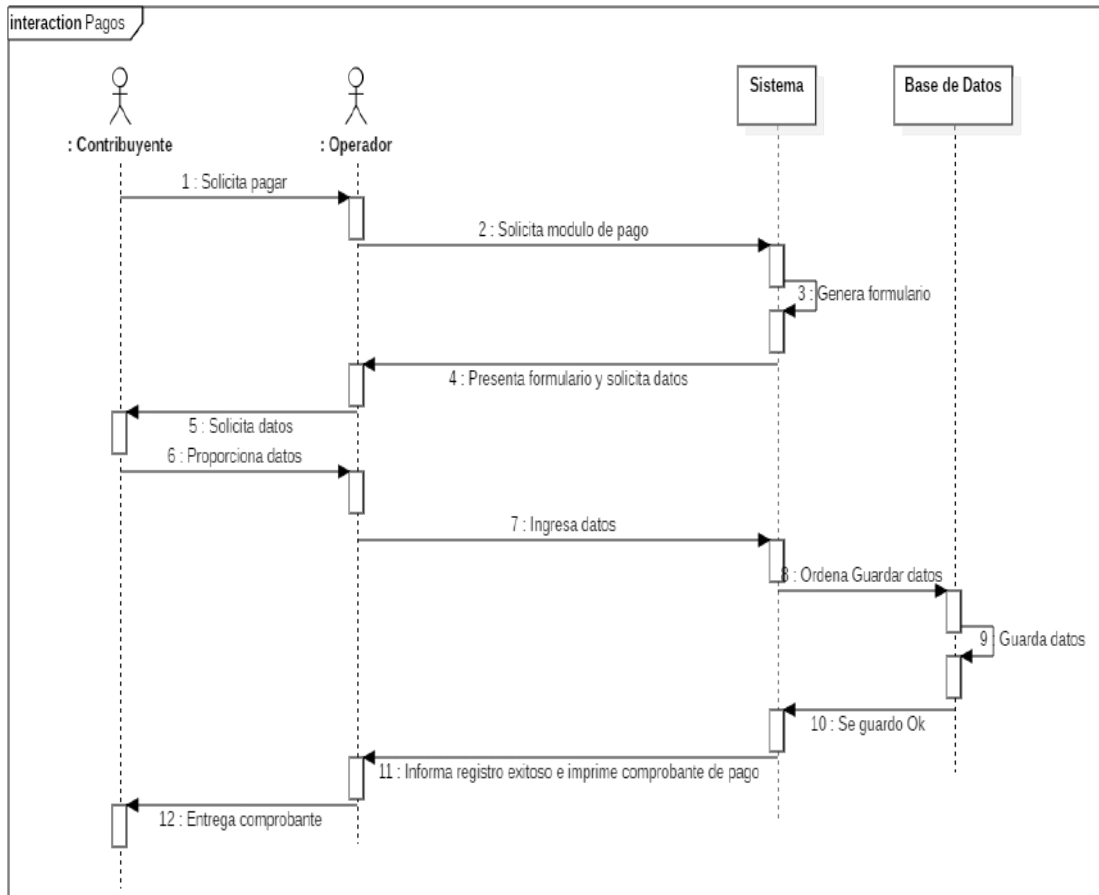
Para generar el monto de contribución el trabajador solicita el módulo generar contribución. El sistema muestra un formulario para llenar información del contribuyente ya registrado anteriormente y costos por obra, finalmente el sistema verifica que todo está correcto y guarda a la base de datos el monto total generado.



**Figura 8.** Diagrama de Secuencia - Registrar Contribuyente y Generar Monto

Fuente: Elaboración Propia

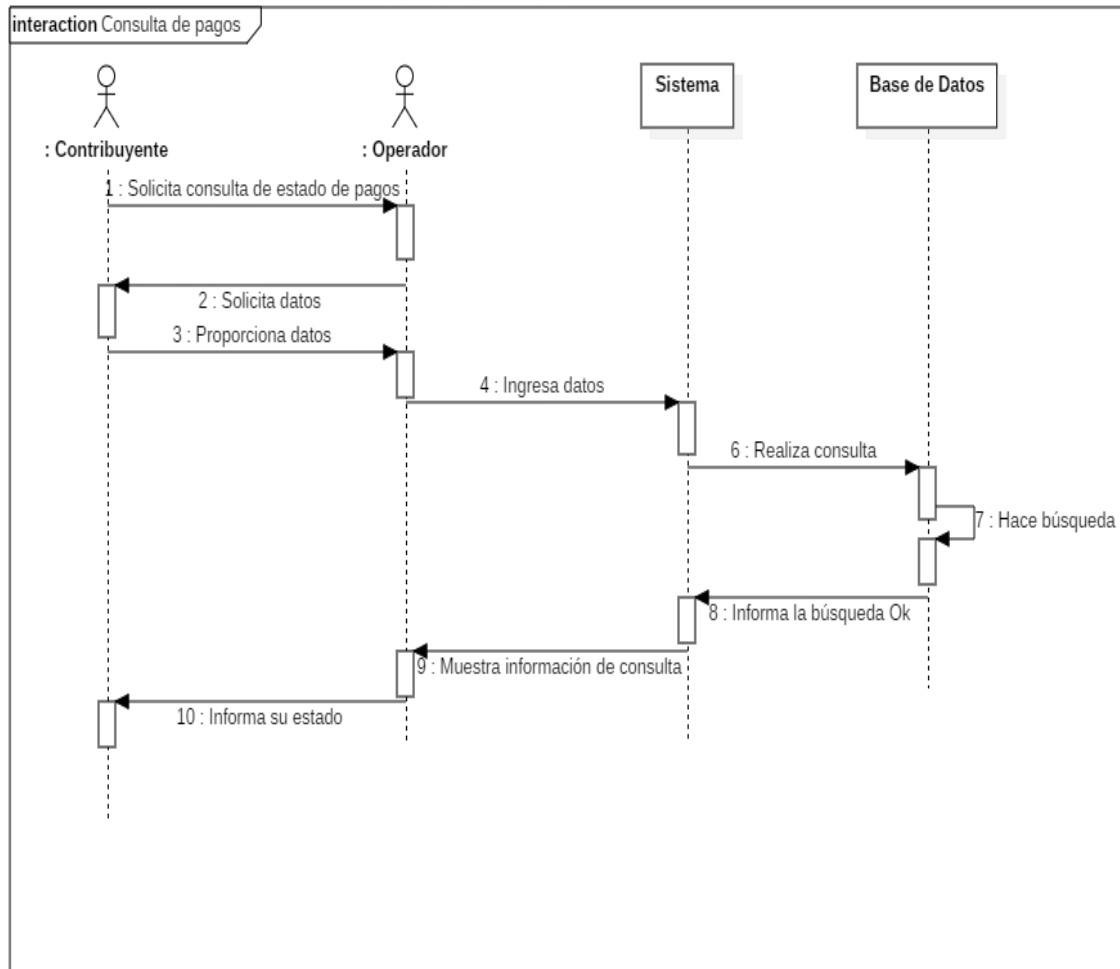
**Interfaz Para Registrar Pagos:** Para realizar el pago por contribución especial de obras públicas el contribuyente comunica la intención al trabajador y esta pregunta por sus datos personales y el monto que pagara y posterior a ello genera el comprobante de pago.



**Figura 9.** Diagrama de secuencia - Registro de Pagos

Fuente: Elaboración Propia

**Interfaz Para Consulta De Pagos:** El contribuyente solicita la consulta del estado de sus pagos, para eso el operador solicita datos al contribuyente para poder consultar a la Base de Datos, para ello la Base de Datos realiza búsqueda e informa al sistema la búsqueda solicitada y finalmente el operador informa la consulta al contribuyente.

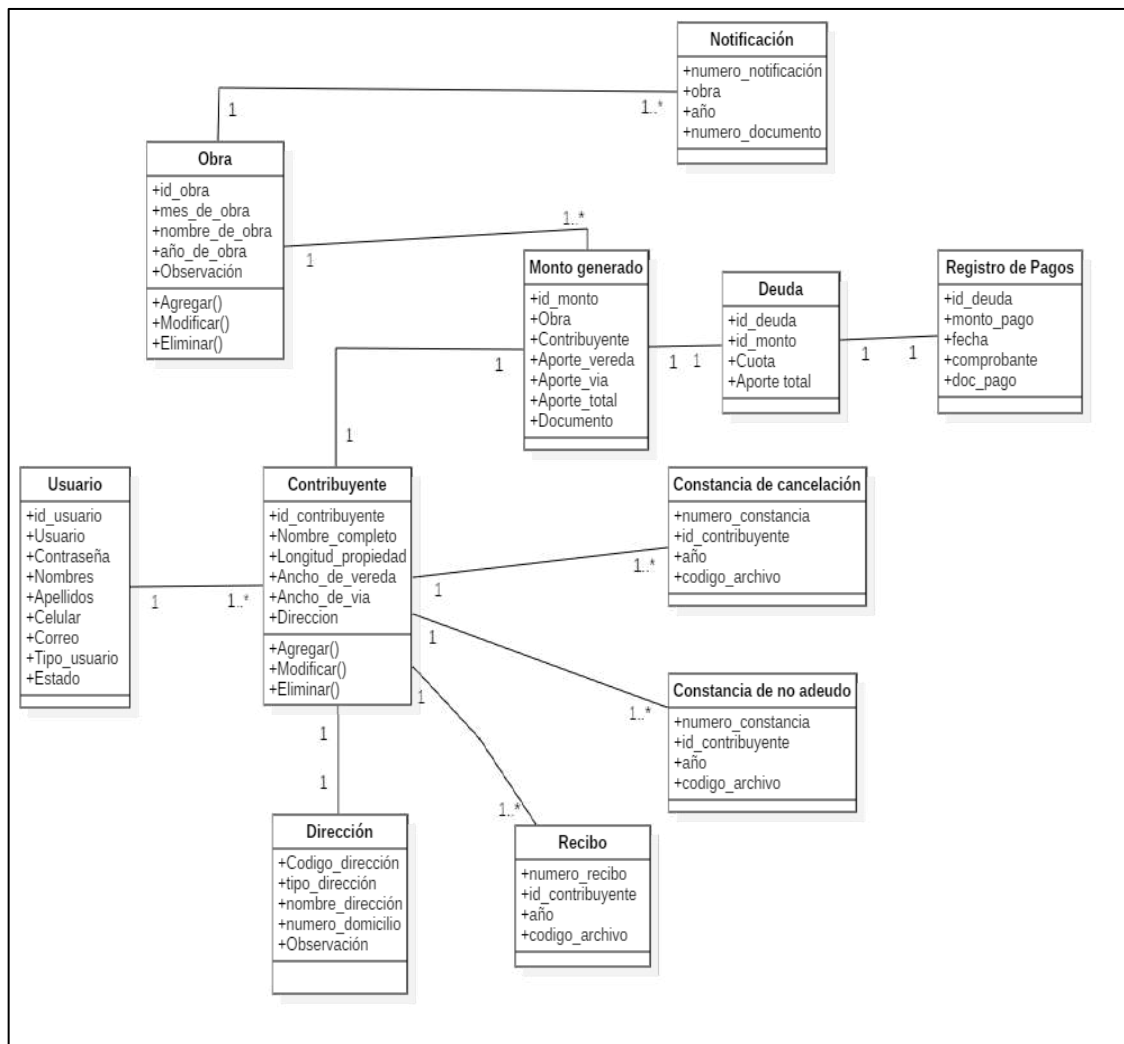


**Figura 10.** Diagrama de secuencia - Consulta de pagos realizados

Fuente: Elaboración Propia

#### 4.2.3 Diagrama de Clases

Este diagrama de clases del sistema proporciona una vista estática del sistema de información desarrollado.



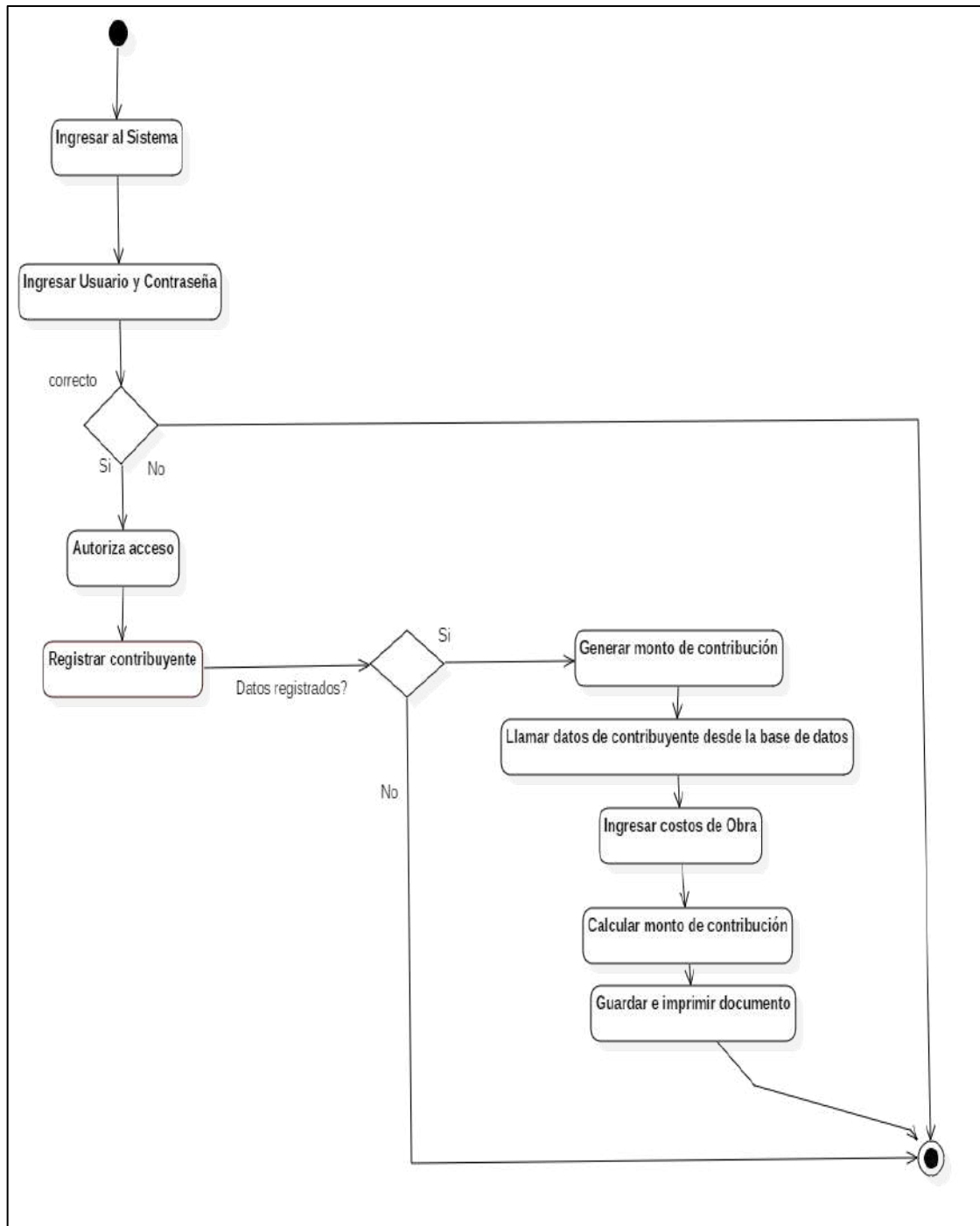
**Figura 11.** Diagrama de Clases

Fuente: Elaboración Propia

#### 4.2.4 Diagrama de Actividades

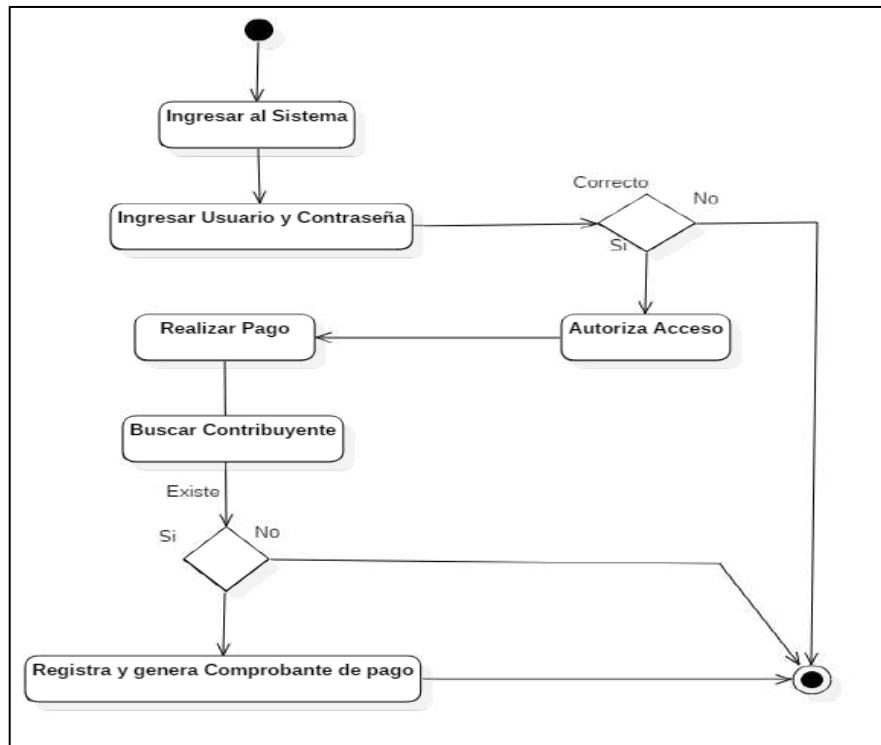
Estos diagramas comprenden básicamente las acciones a ejecutarse en el sistema, en la cual se realiza la representación de las acciones a que realiza el trabajador encargado de manipular el sistema en registro de contribuyentes, generación de monto de contribución, consulta de pagos y emisión de documentos.





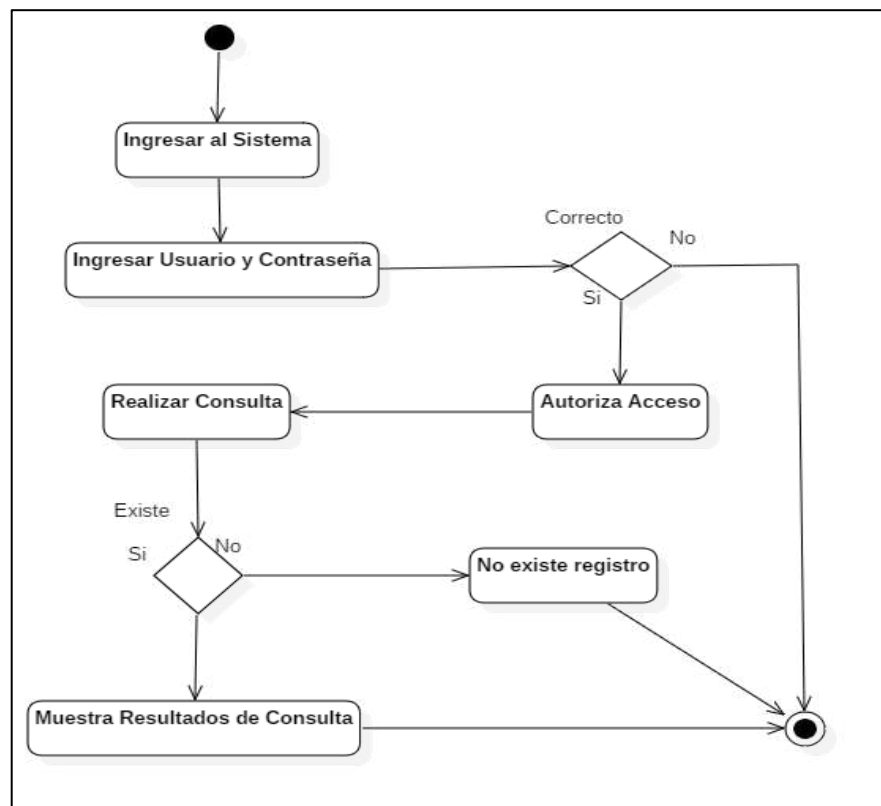
**Figura 12.** Diagrama de Actividad - Registro de Contribuyente

Fuente: Elaboración Propia



**Figura 13.** Diagrama de Actividad - Registrar Pago

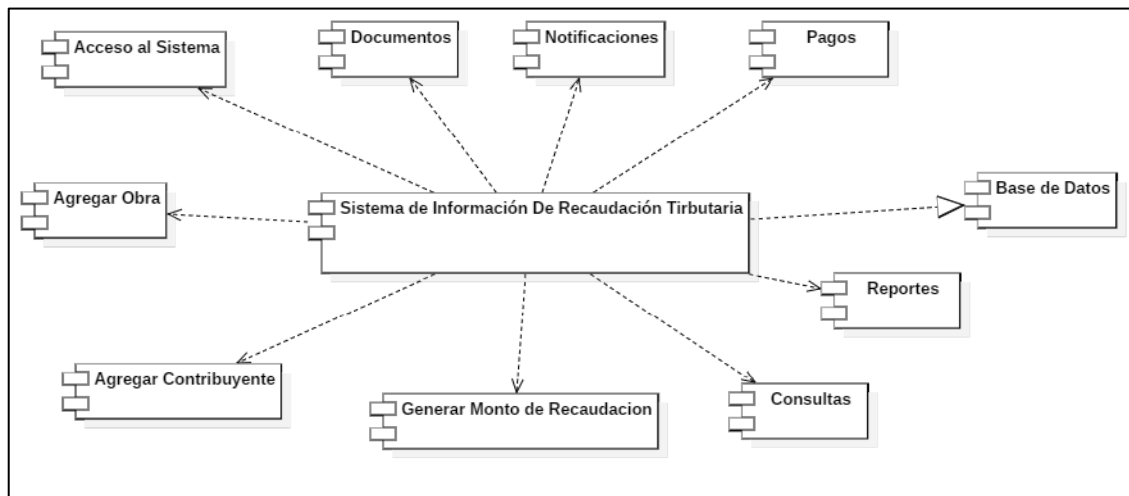
Fuente: Elaboración Propia



**Figura 14.** Diagrama de Actividad - Consultar Pagos

Fuente: Elaboración Propia

#### 4.2.5. Diagrama de Componentes



**Figura 15.** Diagrama de componentes del Sistema de Información

Fuente: Elaboración Propia

#### 4.2.6. Diseño de la base de datos

El diseño de la Base de Datos es fundamental para empezar con el desarrollo de cualquier sistema informático para así asegurar el óptimo funcionamiento. Para el resultado final del diseño se siguió las reglas de Normalización de base de datos llegando hasta la tercera forma normal (ver **anexo D**).

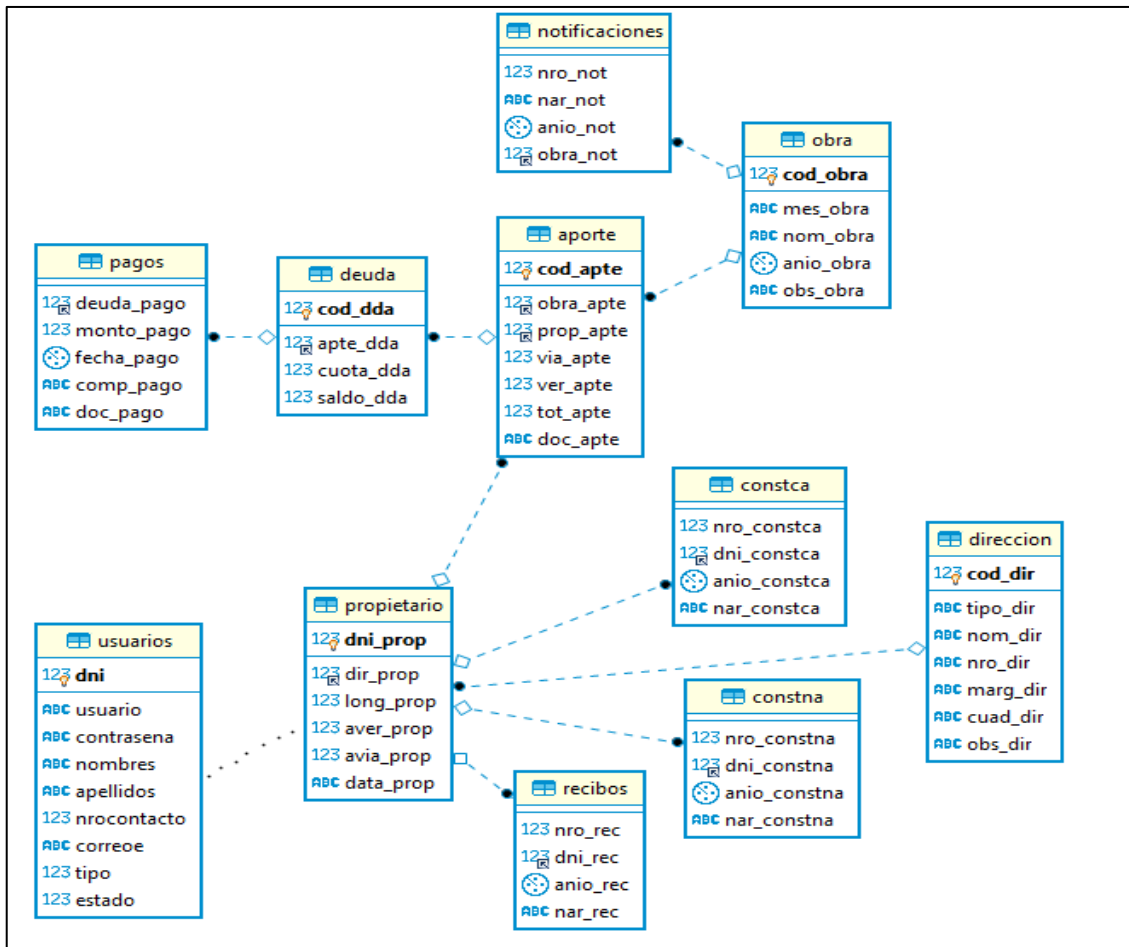


Figura 16. Diseño de Base de Datos

Fuente: Elaboración propia

#### 4.2.7. Arquitectura Física del Sistema

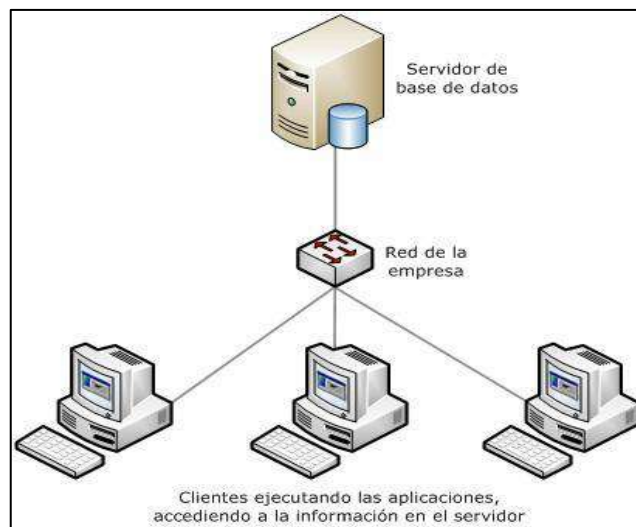
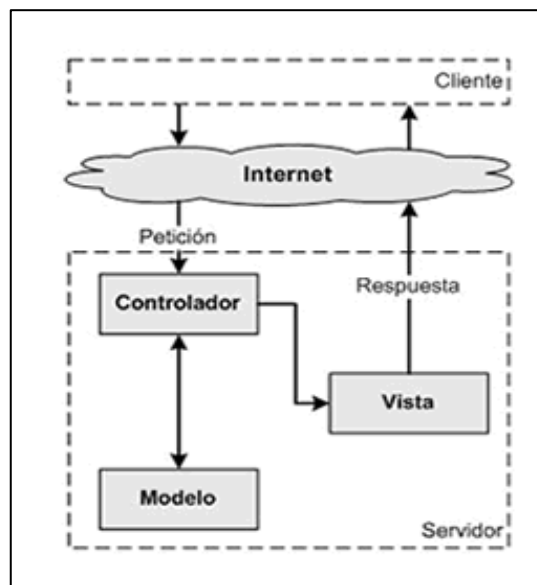


Figura 17. Arquitectura Física del Sistema

Fuente: researchgate.net

#### 4.2.8. Arquitectura Lógica del Sistema

El diseño de la arquitectura lógica del sistema es la siguiente:



**Figura 18.** Arquitectura Lógica del Sistema

Fuente: <http://wwtecnikasoluciones.com/site/sistemas-empresariales/sistemas-a-medida>

#### 4.3. CODIFICACIÓN

La metodología XP recomienda la programación en parejas, sin embargo, el presente proyecto fue desarrollado por una sola persona y no aplicó.

Previo al inicio de la codificación es necesario configurar CodeIgniter, (ver **anexo E**) se detallan los pasos necesarios para instalación y configuración de este framework.

En el (**anexo F**) se muestra la codificación de los módulos del sistema con CodeIgniter en las tres capas MVC.

#### 4.4. MÓDULOS DEL SOFTWARE

En esta fase se muestra el resultado de la implementación del sistema de información bajo el patrón de arquitectura de software Modelo Vista Controlador (MVC), cumpliendo con

los requerimientos funcionales y no funcionales solicitados en la fase de análisis del Sistema. (ver **anexo G**)

#### **4.5. PRUEBA DE SOFTWARE**

Se realizó prueba de caja negra lo cual consistió en la evaluación general de funcionamiento del “SIRPACEPV” y la aceptabilidad por parte de los usuarios ante el Sistema.

Se obtuvo la aprobación del personal que labora en la Sub Gerencia de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno, quienes aprobaron el general funcionamiento del sistema, y afirmaron el impacto positivo que tiene para el manejo de información de las obras de pistas y veredas.

##### **4.5.1. Validación Del Sistema**

Para evaluar la calidad del sistema “SIRPACEPV”, se aplicó fichas de evaluación de ISO – 9126 para medir la calidad de software. El cual fue evaluado por 2 encargados de obras de pistas y veredas (ver **anexo H**).

**Tabla 11.**

*Medición de calidad del software estándar ISO - 9126*

<b>CLASIFICACIÓN</b>	<b>INTERVALO</b>	<b>DECISIÓN</b>
Inaceptable	[ 20 – 38 >	
Mínimamente aceptable	[ 38 – 56 >	
Aceptable	[ 56 – 74 >	
Cumple los requisitos	[ 74 – 92 >	85
Excede los resultados	[ 92 – 110 >	

*Fuente: Estándar ISO - 9126*

### **Decisión:**

La decisión que muestra esta tabla es el resultado del promedio de puntajes que se obtuvieron de las fichas de evaluación, dicho valor es de 85. Lo que indica que el sistema de información “SIRPACEPV” cumple con los requisitos propuestos según el estándar de calidad ISO 9126.

### **Discusión:**

En función del resultado obtenido se afirma que mediante el uso del estándar ISO 9126, incrementa el grado de: funcionalidad, usabilidad, fiabilidad, eficiencia, portabilidad y mantenibilidad. Obteniendo como resultado una puntuación de 86. Lo que cumple con los requisitos según el estándar de calidad ISO 9126. Asimismo se concuerda con los resultados obtenidos de (Chucuya, 2015), quien concluye que el software “SIPRIA”, reduce en el tiempo y son confiables, cumpliendo los requisitos de calidad del producto del software según el estándar ISO 9126.

## **4.6. PRUEBA DE HIPÓTESIS**

Pruebas de hipótesis estadístico de comparación de medias para datos parecidos y cuadro comparativo según el tiempo el antes y el después.

### **i. Planteamiento de hipótesis**

$H_0: \mu_x \geq \mu_y$  (Con la implementación del Sistema de Información con Arquitectura MVC, para la recaudación tributaria por obras públicas de pistas y veredas para la oficina de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno, no se reduce el tiempo de atención a los contribuyentes).

$H_a: \mu_x < \mu_y$  (Con la implementación del Sistema de Información con Arquitectura MVC, para la recaudación tributaria por obras públicas de pistas y veredas para la oficina de Promoción Social y Participación Ciudadana de la

Municipalidad Provincial de Puno, se reduce el tiempo de atención a los contribuyentes).

### ii. Nivel de Significancia:

Nivel de significancia  $\alpha = 0.05 = 5\%$  y un valor tabular  $Z_t$ , en este caso se tiene:

$$Z_t = Z_{0.05} = 1.645$$

### iii. Prueba Estadística:

$$Z_c = \frac{\bar{d}}{S_d/\sqrt{n}} = \frac{-10.78}{9.76/\sqrt{66}} = -8.99$$

$$Z_c = |-8.99|$$

$$Z_c = 8.99$$

### iv. Decisión

como  $Z_c > Z_t$  ( $Z_c$  calculada es mayor a  $Z_t$  tabular), entonces se rechaza la  $H_0$  y se acepta la  $H_a$ .

**Interpretación:** Lo que nos indica que tiempo promedio de atención a los contribuyentes después de implementar el Sistema de Información es menor al tiempo promedio de atención antes, es decir hay una diferencia significativa y que con el uso de “SIRPACEPV” se reduce el tiempo de atención.



## RESULTADOS DEL CUESTIONARIO

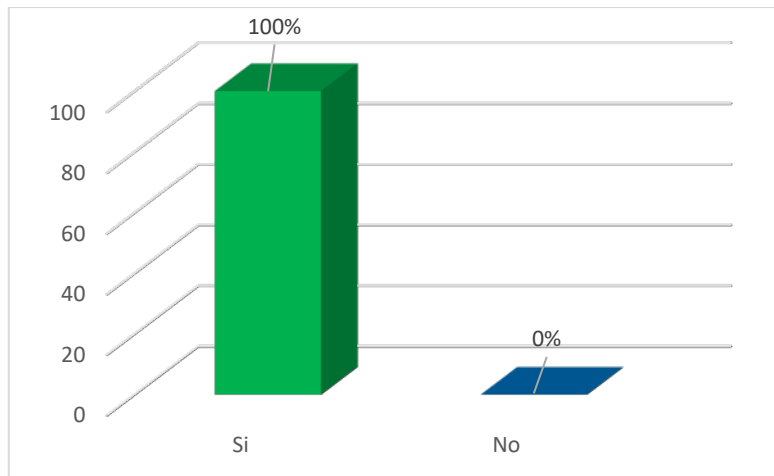
### Resultados Sobre la Implementación Del Sistema

**Tabla 12.**

*La implementación del sistema "SIRPACEPV" mejora la gestión administrativa*

<b>El Sistema Mejora La Gestión Administrativa</b>	<b>Frecuencia</b>	<b>Porcentaje (%)</b>
Si	5	100
No	0	0
<b>Total</b>	<b>5</b>	<b>0</b>

*Fuente: Elaboración Propia*



**Figura 19.** Implementación de "SIRPACEPV" mejora la Gestión Administrativa

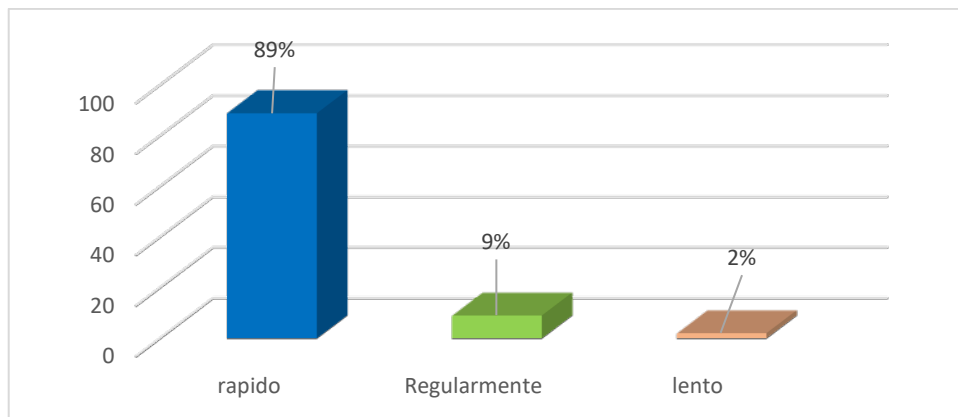
Fuente: Elaboración Propia

Se observa que el 100% del personal que labora en la subgerencia de promoción social y participación ciudadana de la Municipalidad Provincial de Puno, considera que la implementación del sistema "SIRPACEV" si mejoró la gestión administrativa.

**Tabla 13.** *Tiempo de atención con el uso del sistema "SIRPACEPV".*

Tiempo De Atención	Frecuencia	Porcentaje (%)
Rápido	59	89
Regularmente	6	9
Lento	1	2
<b>Total</b>	<b>66</b>	<b>100</b>

*Fuente: Elaboración Propia*



**Figura 20.** Tiempo de atención con el uso del sistema "SIRPACEPV"

Fuente: Elaboración Propia

De acuerdo a la encuesta aplicada en la subgerencia de promoción social y participación ciudadana de la Municipalidad Provincial de Puno, se observa que el 89% de los contribuyentes considera que el tiempo de atención con el uso del sistema "SIRPACEPV" es rápido mientras que el 9% de los contribuyentes considera que es regularmente y el 2% considera que es lento.

#### 4.7. DISCUSIÓN

De acuerdo con los resultados obtenidos se afirma que la implementación de un Sistema de Información con arquitectura MVC para la recaudación tributaria por obras públicas de la Municipalidad Provincial de Puno, permitió mejorar el tiempo de atención



a los contribuyentes mediante la automatización de procesos y tareas de recaudación que realiza el personal encargado. Este resultado se refuerza con el trabajo de (Valdivia, 2018) en su investigación de **aplicación web basada en el patrón MVC para mejorar la gestión tributaria de la municipalidad distrital de Guadalupito, la libertad** cuyo objetivo es mejorar significativamente la gestión tributaria de la Municipalidad Distrital de Guadalupito con el desarrollo de una aplicación web, así como reducir el tiempo en la búsqueda de información de los contribuyentes, que finalmente concluye que la aplicación web, en la municipalidad Distrital de Guadalupito, influye en gran escala, puesto que permite reducir el tiempo en la búsqueda de información de los contribuyentes. Así mismo se concuerda con los resultados de (Chuyes & Carreño, 2014), quien concluye que se ha desarrollado un sistema que satisface completamente las necesidades del Usuario (Área de Rentas) dando así una herramienta optima y benéfica para la Municipalidad Distrital de Bellavista para que pueda mejorar los procesos de recaudación Tributaria. Estos resultados se relacionan con la investigación de (Kawakami, 2016) quien presenta una **propuesta de un sistema de recaudación tributaria del impuesto predial para optimizar la gestión pública de la Municipalidad Provincial de Bagua Amazonas-2016**, llegando a la conclusión que se debería implantar un Sistema de Recaudación Tributaria ya que esto permitiría una adecuada Gestión y procesamiento de datos, logrando ser más eficiente el servicio. De la misma manera (Tineo & Aguilar, 2018), en su trabajo de investigación de **Propuesta de uso de Gobierno Electrónico para mejorar la Recaudación Tributaria en la Municipalidad Distrital de Pomalca, 2017**, concluye que los problemas de recaudación tributaria viene en un 40 % por la mala recaudación tributaria y que un **80%** por carecer de recursos tecnológicos y equipos informáticos.



## V. CONCLUSIONES

Una vez terminada la investigación, acerca de la implementación de un sistema de información con arquitectura MVC para la recaudación tributaria de obras públicas de pistas y veredas de la Municipalidad Provincial de Puno, de acuerdo a los resultados obtenidos se llega a las siguientes conclusiones:

**PRIMERA:** El Sistema de Información desarrollado en el presente trabajo de investigación tiene buenos tiempos de respuesta debido a que se utilizó AJAX, que es una técnica de desarrollo web basada en la tecnología JavaScript que permite intercambiar datos con el servidor en segundo plano sin necesidad de recargar toda la página.

**SEGUNDA:** La aplicación del patrón de diseño MVC mediante el framework CodeIgniter permitió la separación clara de donde se ubica cada tipo de lógica de manera ordenada, facilitando el mantenimiento y la escalabilidad del sistema de información.

**TERCERA:** La combinación del patrón MVC, framework CodeIgniter y la metodología ágil XP, fueron claves para la implementación de SIRPACEPV, fundamentalmente por la facilidad de dar soluciones rápidas y de calidad en los cambios que se presentaron durante la implementación.

**CUARTA:** El framework CodeIgniter garantiza un alto grado de seguridad en el sistema implementado por que usa sesiones encriptadas que protegen la lectura de su contenido, filtro de seguridad XSS (Cross Site Scripting) en todos los datos enviados por medio de formularios y finalmente por la preferencia de uso del formulario POST en lugar de GET.



**QUINTA:** La implementación del sistema de información para la recaudación tributaria de pistas y veredas de la Municipalidad Provincial de Puno, permitió mejorar de manera rápida y oportuna la atención a los contribuyentes tal como se demostró en el resultado de la prueba estadística de comparación de medias para datos apareados,  $Z_c(8.99) > Z_t(1.645)$  lo que nos indica que el tiempo promedio de atención es mucho mejor, gracias a que el sistema permite al personal realizar los procesos de manera automatizada y eficiente.



## VI. RECOMENDACIONES

- PRIMERA:** Se recomienda mantener actualizado el registro de contribuyentes en el Sistema puesto que esta información es utilizada en la mayoría de los procesos que se realizan en el proceso de contribución por obras públicas.
- SEGUNDA:** Realizar copias de respaldo de la base de datos de forma periódica.
- TERCERA:** El mantenimiento del sistema debe ser por un personal con sólidos conocimientos de programación web y base de datos.
- CUARTA:** Se recomienda a futuras investigaciones implementar una aplicación móvil que permita hacer consultas a los contribuyentes vía “On - line” de sus pagos realizados y deuda pendiente por contribución especial de obras públicas de pistas y veredas.



## VII. REFERENCIAS BIBLIOGRÁFICAS

- Abud, M. A. (2012). *Calidad en la Industria del Software. La Norma ISO-9126*. 2–4.  
<http://148.204.210.204/revistaupiicsa/34/34-2.pdf>
- Auladell, G. (2016). *¿Qué es MariaDB? | Drauta*. <https://www.drauta.com/que-es-mariadb>
- Ayoze Castillo, A. (2017). *Curso de Programación Web: JavaScript, Ajax y jQuery* (I. C. Academy (ed.); 2ª Edición).
- Barra, J. A., & Limache, M. (2017). *Sistema de Gestión para el control de clientes morosos en el Banco “mibanco” del periodo 2015*. [Universidad Nacional del Altiplano]. <http://repositorio.unap.edu.pe/handle/UNAP/7088>
- Chucuya, V. (2015). *Implementación de un sistema de información para la recaudación del impuesto de alcabala ‘SIPRIA’ de la municipalidad provincial de puno - 2015*. [Universidad Nacional del Altiplano].  
<http://repositorio.unap.edu.pe/handle/UNAP/1945>
- Chuyes, Y. M. B., & Carreño, M. I. (2014). *Desarrollo e implementación del sistema de rentas de la municipalidad distrital de bella vista*. [UNIVERSIDAD NACIONAL DE PIURA]. <http://repositorio.unp.edu.pe/handle/UNP/648>
- CodeIgniter. (2018). *Manual CodeIgniter en español*. <https://codeigniter.com>
- Cordova, M. (2003). *Estadística descriptiva e inferencial* (Quinta Edi).
- Cortés, R. (2005). *Introducción al análisis de sistemas y la ingeniería de Software* (Primera). Editorial Universidad Estatal a Distancia.
- De los Rios, H. M. (2015). *Modelo de adecuación y arquitectura de aplicaciones utilizando patrones arquitectónicos modelo vista controlador en el frontend*. [UNIVERSIDAD COMPLUTENSE DE MADRID]. <https://eprints.ucm.es/32809/>



- Fernández, V. (2010). *Desarrollo de Sistemas de Información: Una Metodología Basada en el Modelado* (1ª. Ed). Univ. Politec. de Catalunya.  
[https://books.google.com.pe/books?id=Sqm7jNZS\\_L0C&hl=es&source=gbs\\_navlinks\\_s](https://books.google.com.pe/books?id=Sqm7jNZS_L0C&hl=es&source=gbs_navlinks_s)
- Gallardo, R. C. (2018). *Desarrollo de un sistema para gestionar órdenes de un restaurante, usando mvc*. [ONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR]. <http://repositorio.puce.edu.ec/handle/22000/15842>
- García, M. (2017). *MVC (Modelo-Vista-Controlador): ¿qué es y para qué sirve?*  
<https://codingornot.com/mvc-modelo-vista-controlador-que-es-y-para-que-sirve>
- Gauchat, J. . (2017). *El gran libro de HTML5, CSS3 y JavaScript* (3a edición). MARCOMBO, S.A. <https://elevaciondigital.pe/wp-content/uploads/2019/06/El-gran-libro-de-HTML5-CSS3-y-JavaScript.pdf>
- Hernández Sampieri, R., Méndez Valencia, S., & Menodoza Torres, C. P. (2017). Fundamentos de investigación. In S. A. D. C. . McGRAW-HILL/INTERAMERICANA EDITORES (Ed.), *Fundamentos de investigación* (Primera ed). <https://doi.org/10.33132/9789585459670>
- Heurtel, O. (2016). *PHP 7 Desarrollar un sitio web dinamico e interactivo*. Ediciones ENI.
- Kawakami, J. L. (2016). *Propuesta de un sistema de recaudación tributaria del impuesto predial para optimizar la gestión publica de la municipalidad provincial de bagua - año 2016*. [Universidad Cesar Vallejo].  
<http://repositorio.ucv.edu.pe/handle/UCV/10176>
- nginx. (2017). *Bienvenido a NGINX Wiki*. <https://www.nginx.com/resources/wiki/>
- Pressman, R. S. (2010). *ingeniería del software Un enfoque práctico* (Séptima ed). McGRAW-HILL. [http://artemisa.unicauca.edu.co/~cardila/Libro\\_Pressman\\_7.pdf](http://artemisa.unicauca.edu.co/~cardila/Libro_Pressman_7.pdf)





- Salvador, L. (2018). *Modelos de datos y visión conceptual de una base de datos*.  
Editorial Elearning, S.L.  
<https://books.google.com.pe/books?id=LV9WDwAAQBAJ>
- Tapuy, S. S. (2019). *Sistema de información con arquitectura mvc para el control de inventario de productos de la empresa “decorey puyo*. [UNIVERSIDAD REGIONAL AUTÓNOMA DE LOS ANDES].  
<http://dspace.uniandes.edu.ec/handle/123456789/9820>
- Tineo, E., & Aguilar, A. (2018). Propuesta de uso de Gobierno Electrónico para mejorar la Recaudación Tributaria en la Municipalidad Distrital de Pomalca, 2017. [Universidad César Vallejo]. In *Universidad César Vallejo*.  
<http://repositorio.ucv.edu.pe/handle/UCV/29194>
- Valdivia, M. S. (2018). *Aplicación web basada en el patrón mvc para mejorar la gestión tributaria de la municipalidad distrital de guadalupito, la libertad*. [Universidad César Vallejo]. <http://repositorio.ucv.edu.pe/handle/UCV/36572>
- Vega, J. F., & Van Der Henst, C. (2011). *Guía HTML 5*.  
<https://openlibra.com/es/book/guia-html5-el-presente-de-la-web>



## ANEXOS



**Anexo A: Encuesta sobre la implementación del sistema de información mejora la gestión administrativa de recaudación tributaria por obras públicas.**



**UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO**  
**FACULTAD DE INGENIERIA ESTADÍSTICA E INFORMÁTICA**



**ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA**

La presente encuesta se realiza con un fin investigativo, se ruega pueda brindar la información de la manera más apropiada, agradecemos enormemente el tiempo y la importancia que se presta para poder llevar a cabo la recopilación de la información necesaria para desarrollar esta investigación. Se le pide que llene o marque según corresponda las siguientes preguntas:

1.- ¿Cómo considera Ud. el servicio que ofrece la oficina de Sub Gerencia de Promoción Social y Participación Ciudadana de la Municipalidad Provincial de Puno en la recaudación tributaria por obras públicas de pistas y veredas?

- a) Deficiente
- b) Regularmente
- c) Eficientes

2.- ¿Cree Ud. que, con el Diseño e implementación de un Sistema de información para la recaudación tributaria por obras públicas de pistas y vereda, mejore la gestión administrativa?

- a) Si.
- b) No.

3.- ¿Cuál es el tiempo (en minutos) que se demoran en realizar el pago por contribución especial por obras públicas de pistas y veredas?

.....



## Anexo B: Encuesta a los trabajadores y funcionamiento de SIRPACEPV



UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO  
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



### ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

---

La presente encuesta se realiza con un fin investigativo, se ruega pueda brindar la información de la manera apropiada, agradecemos enormemente el tiempo y la importancia que se presta para poder llevar a cabo la recopilación de la información necesaria para desarrollar esta investigación. Se le pide que llene o marque según corresponda las siguientes preguntas: sobre el sistema de información para la recaudación tributaria por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno

1. ¿Cómo considera Ud. la Implementación de “SIRPACEPV”?
  - a) Muy bueno
  - b) Bueno
  - c) Regular
  
2. ¿Cómo considera Ud. la interacción con el sistema para la recaudación tributaria al registrar?
  - a) Muy Fácil
  - b) Fácil
  - c) Difícil
  
3. ¿Cómo considera Ud. los servicios que ofrece el sistema implementado para la Oficina De Promoción Social Y Participación Ciudadana?
  - a). Eficiente
  - b) Regularmente
  - c) Deficiente
  
4. ¿El ingreso de los datos y/o registros al sistema es?
  - a) Muy Fácil
  - b) Fácil
  - c) Complicado
  
5. ¿Cree Ud. que el registro de pagos por contribución especial por obras públicas de pistas y veredas son confiables?
  - a) Si
  - b) No
  
6. ¿Considera Ud. que los reportes que emite el sistema es satisfactorio?
  - a) Muy Bueno
  - b) Bueno
  - c) Regular



## Anexo C: Encuesta antes y después de implementar SIRPACEPV



UNIVERSIDAD NACIONAL DEL ALTIPLANO – PUNO  
FACULTAD DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA



### ESCUELA PROFESIONAL DE INGENIERÍA ESTADÍSTICA E INFORMÁTICA

---

La presente encuesta se realiza con un fin investigativo, se ruega pueda brindar la información de la manera más apropiada, agradecemos enormemente el tiempo y la importancia que se presta para poder llevar a cabo la recopilación de la información necesaria para desarrollar esta investigación. Se le pide que llene o marque según corresponda las siguientes preguntas: sobre el sistema de información para la recaudación tributaria por obras públicas de pistas y veredas de la Municipalidad Provincial de Puno

1. ¿Cómo considera el tiempo de atención con el uso del sistema para el pago de contribución especial por obras públicas de pistas y veredas?
  - a) Rápido
  - b) Regularmente
  - c) Lento
2. ¿Teniendo en cuenta el tiempo en el proceso de registro que antiguamente se realizaba y con la actual implementación del sistema para el pago de contribución especial por obras públicas de pistas y veredas, cuál de los dos Ud. sugiere que se utilice?
  - a). El uso del sistema.
  - b) El método tradicional
  - c) Ambos
3. ¿Cuál era el tiempo que se demoraba para realizar el pago de contribución especial por obras públicas de pistas y veredas antes de la implementación del sistema?

.....
4. ¿Cuál es el tiempo que se demora para realizar el pago de contribución especial por obras públicas de pistas y veredas con la implementación del sistema?

.....



## Anexo D: Normalización de Base de Datos.

### TABLA NO NORMALIZADA

ID	Nombre Propietario	Dirección	Longitud de la propiedad	Área de vereda	Área de Vía	Aporte Generado	Deuda	constancia obra	pagos	recibo
----	--------------------	-----------	--------------------------	----------------	-------------	-----------------	-------	-----------------	-------	--------

### PRIMERA FORMA NORMAL

ID	Nombre propietario	Longitud de la propiedad	Área de vereda	Área de Vía	Aporte generado	Deuda	Constancia	Pagos	recibo
----	--------------------	--------------------------	----------------	-------------	-----------------	-------	------------	-------	--------

### Tabla dirección

ID	Tipo de dirección	Nombre dirección	Numero de dirección
1	JIRON	CHAVIN	MZ B LOTE 3
2	JIRON	CHAVIN	B-4

### Tabla Obra

Id obra	Mes de la obra	Nombre de la obra	Año de ejecución	observaciones
1	Octubre	OBRA DE PAVIMENTO DE VÍA Y VEREDA DE JIRÓN CHAVIN	2019	PORTE VECINAL POR M2 VIA: S/. 25.00 NUEVOS SOLES

### SEGUNDA FORMA NORMAL

#### Tabla Propietario

ID	Nombre propietario	Longitud de la propiedad	Área de vereda	Área de Vía	Constancia	Pagos
----	--------------------	--------------------------	----------------	-------------	------------	-------

#### Tabla dirección

ID	Tipo de dirección	Nombre dirección	Numero de dirección
1	JIRON	CHAVIN	MZ B LOTE 3
2	JIRON	CHAVIN	B-4



### Tabla Obra

Id obra	Mes de la obra	Nombre de la obra	Año de ejecución	observaciones
1	Octubre	OBRA DE PAVIMENTO DE VÍA Y VEREDA DE JIRÓN CHAVIN	2019	PORTE VECINAL POR M2 VIA: S/. 25.00 NUEVOS SOLES.

### Tabla Aporte

Id aporte	Id obra	Id propietario	Aporte de vía	Aporte de vereda	Aporte total	Documento del aporte
1	1	1	25	15	692	b06e3dbb77608055b1c6e1d2e9c565c.pdf
2	1	2	25	15	684	b06e3dbb77608055b1c6e1d2e9c565c.pdf
3	1	3	25	15	793	b06e3dbb77608055b1c6e1d2e9c565c.pdf

### Tabla Recibos

Id recibo	Id propietario	Año del recibo	Documento
1	1	2020	b06e3dbb77608055b1c6e1d2e9c565c.pdf
2	2	2020	b06e3dbb77608055b1c6e1d2e9c565c.pdf

### Tabla Deuda

ID	Id aporte	Aporte 40%	Saldo
1	1	277	277

## TERCERA FORMA NORMAL

### Tabla Propietario

ID	Nombre propietario	Longitud de la propiedad	Área de vereda	Área de Vía
1	Juan Carlos	6	3.5	4
2	María	5	2	3

### Tabla Dirección

ID	Tipo de dirección	Nombre dirección	Numero de dirección
1	JIRON	CHAVIN	MZ B LOTE 3
2	JIRON	CHAVIN	B-4

### Tabla Obra

Id obra	Mes de la obra	Nombre de la obra	Año de ejecución	observaciones
1	Octubre	OBRA DE PAVIMENTO DE VÍA Y VEREDA DE JIRÓN CHAVIN	2019	PORTE VECINAL POR M2 VIA: S/. 25.00 NUEVOS SOLES



### Tabla Aporte

<b>Id aporte</b>	<b>Id obra</b>	<b>Id propietario</b>	<b>Aporte de vía</b>	<b>Aporte de vereda</b>	<b>Aporte total</b>	<b>Documento del aporte</b>
1	1	1	25	15	692	b06e3dbb77608055b1c6e1d2e9c565c.pdf
2	1	2	25	15	684	b06e3dbb77608055b1c6e1d2e9c565c.pdf

### Tabla Recibo

<b>Id recibo</b>	<b>Id propietario</b>	<b>Año del recibo</b>	<b>Documento</b>
1	1	2020	b06e3dbb77608055b1c6e1d2e9c565c.pdf
2	2	2020	b06e3dbb77608055b1c6e1d2e9c565c.pdf

### Tabla Deuda

<b>ID</b>	<b>Id aporte</b>	<b>Aporte 40%</b>	<b>Saldo</b>
1	1	277	277

### Tabla Pagos

<b>Id deuda</b>	<b>Monto de pago</b>	<b>Fecha de pago</b>	<b>Numero de comprobante</b>	<b>Documento</b>
1	277	2019-04-22	001176922019	b06e3dbb77608055b1c6e1d2e9c565c.jpg

### Tabla Constancias

<b>Id constancia</b>	<b>Id propietario</b>	<b>Año</b>	<b>Documento</b>
1	1	2020	b06e3dbb77608055b1c6e1d2e9c565c.jpg





## Anexo E: Instalación y configuración de CodeIgniter

### INSTALACIÓN Y CONFIGURACIÓN DE CODEIGNITER

Los pasos necesarios para instalar y configurar CodeIgniter son los siguientes:

1. Instalar XAMPP.
2. Descargar CodeIgniter de la url <https://codeigniter.com/download>.
3. Descomprimir la carpeta y renombrar.
4. Copiar la carpeta en htdocs ubicada en C:\xampp\.
5. Crear la base de datos en MariaDB.
6. Configurar los archivos necesarios de CodeIgniter.

Configurar el archivo *autoload.php*

*C:\xampp\htdocs\application\config*

---

```
$autoload['libraries'] = array('database','session','form_validation','upload');  
// librerias para conexion base de datos, sesiones, validaciones,  
y carga de archivos.
```

---

---

```
$autoload['helper'] = array('security','url','form','language');  
// Se carga la libreria para la url, formularios y seguridad.
```

---

Configurar el archivo *config.php*

*C:\xampp\htdocs\application\config*

---

```
$config['base_url'] = ((isset($_SERVER['HTTPS']) && $_SERVER['HTTPS'] == "on") ? "https" : "http");  
$config['base_url'] .= "://".$_SERVER['HTTP_HOST'];  
$config['base_url'] .= str_replace(basename($_SERVER['SCRIPT_NAME']), "", $_SERVER['SCRIPT_NAME']);
```

---

```
$config['global_xss_filtering'] = TRUE;  
// filtro de prevención de Cross Site Scripting Hack
```

---



Configurar el archivo *database.php*

*C:\xampp\htdocs\application\config\database.php*

---

```
'hostname' => 'localhost', //nombre del hostname.  
'username' => 'root', //usuario de la base de datos.  
'password' => '', //contraseña de la base de datos  
'database' => 'sirpacepv', //nombre de la base de datos  
'dbdriver' => 'mysqli',  
'dbprefix' => '',
```

---

Configurar el archivo *routes.php*

*C:\xampp\htdocs\application\config\routes.php*

---

```
$route['default_controller'] = 'sacppv/ingresar'; //controlador por  
default  
$route['translate_uri_dashes'] = FALSE;  
$route['404_override'] = 'sacppv/error404'; //controlador para error  
es
```

---



## Anexo F: Estructura del código fuente basado en Modelo Vista Controlador.

\*\*\*\*\*

\*Sistema de registro de pagos por contribución especial por obras públicas.

\*Versión 1.0

\*Programador: Bach Juan Carlos Huanca Mollinedo.

\*\*\*\*\*

### CONTROLADOR

---

```
<?php
class Sacppv extends CI_Controller
{
    public function __construct()
    {
        parent::__construct();
        $this->load-
>model('Autenticacion_model', 'autenticacion');
        $this->load->model('Obra_model', 'obra');
        $this->load->model('Propietario_model', 'propietario');
        $this->load->model('Aporte_model', 'aporte');
    }
    public function ingresar()
    {
        if ($this->session-
>userdata('ci_session_key_generate') == true) {
            $this->form_validation-
>set_rules('usuario', 'Nombre de usuario', 'trim|required');

            $this->form_validation-
>set_rules('contrasena', 'Contraseña', 'trim|required');

            if ($this->form_validation->run() == false) {

                //Field validation failed. User redirected to login page
                $this->load->view('pages/login');
            } else {
                $sessArray = array();
                //Field validation succeeded. Validate against d
                atabase
            }
        }
    }
}
```

---



```
$usuario = $this->input->post('usuario');
$contrasena = $this->input->post('contrasena');

$this->autenticacion->setusuario($usuario);
$this->autenticacion-
>setcontrasena($contrasena);
//query the database
$result = $this->autenticacion->login();
}

public function obras()
{
    if ($this->session-
>userdata('ci_session_key_generate') == false) {
        redirect('ingresar');
    } elseif ($this->session-
>userdata('ci_seesion_key')['tipo'] == 1) {
        $data['titulo'] = "Obras";
        $this->load->view('templates/header', $data);
        $this->load->view('pages/obras/index');
        $this->load->view('templates/footer');
    } else {
        redirect('error403');
    }
}

public function propietarios()
{
    if ($this->session-
>userdata('ci_session_key_generate') == false) {
        redirect('ingresar');
    } elseif ($this->session-
>userdata('ci_seesion_key')['tipo'] == 1) {
        $data['titulo'] = "Propietarios";
        $this->load->view('templates/header', $data);
        $this->load->view('pages/propietarios/index');
        $this->load->view('templates/footer');
    } else {
        redirect('error403');
    }
}

public function generar_aporte()
{
    if ($this->session-
>userdata('ci_session_key_generate') == false) {
        redirect('ingresar');
    } elseif ($this->session-
>userdata('ci_seesion_key')['tipo'] == 1) {
        $data['titulo'] = "Aporte";
```

```
        $this->load->view('templates/header', $data);
        $this->load->view('pages/costo');
        $this->load->view('templates/footer');
    } else {
        redirect('error403');
    }
}

public function pagar_aporte()
{
    if ($this->session-
>userdata('ci_session_key_generate') == false) {
        redirect('ingresar');
    } elseif ($this->session-
>userdata('ci_seesion_key')['tipo'] == 1) {
        $data['titulo'] = "Pagar";
        $this->load->view('templates/header', $data);
        $this->load->view('pages/pagar');
        $this->load->view('templates/footer');
    } else {
        redirect('error403');
    }
}

public function guardaraporte()
{
    if ($this->session-
>userdata('ci_session_key_generate') == false) {
        redirect('ingresar');
    } elseif ($this->session-
>userdata('ci_seesion_key')['tipo'] == 1) {
        $json = array();
        $sobrapte = $this->input->post('idObra');
        $propappte = $this->input->post('idpropietario');
        $verappte = $this->input->post('costoVereda');
        $viaappte = $this->input->post('CostoVia');
        $totappte = $this->input->post('aporteTotal');

        if (empty(trim($propappte))) {
            $json['error']['idpr'] = 'Falta datos del propiet
ario';
        }
        if ($verappte < 0) {
            $json['error']['cosver'] = 'Falta costo de vereda
';
        }
        if ($viaappte < 0) {
            $json['error']['cosvia'] = 'Falta costo de vía';
        }
        if (empty(trim($totappte))) {
```



```
        $json['error']['aptot'] = 'Falta costo total gene  
rado';  
    }  
    if (empty(trim($obraapte))) {  
        $json['error']['obraapte'] = 'Falta obra';  
    }  
    if ($this->aporte->buscar_aporte($propapte) != []) {  
        $json['error']['idpr'] = 'ID ' . $propapte . ' ya  
generó un aporte';  
    }  
  
    public function reporte_pagos()  
    {  
        if ($this->session-  
>userdata('ci_session_key_generate') == false) {  
            redirect('ingresar');  
        } elseif ($this->session-  
>userdata('ci_session_key')['tipo'] == 1 || $this->session-  
>userdata('ci_session_key')['tipo'] == 2) {  
            $data['titulo'] = "Reporte";  
            $this->load->view('templates/header', $data);  
            $this->load->view('pages/reportepagos');  
            $this->load->view('templates/footer');  
        } else {  
            redirect('error403');  
        }  
    }  
  
    public function notificar_pagos()  
    {  
        if ($this->session-  
>userdata('ci_session_key_generate') == false) {  
            redirect('ingresar');  
        } elseif ($this->session-  
>userdata('ci_session_key')['tipo'] == 1) {  
            $data['titulo'] = "Notificar";  
            $this->load->view('templates/header', $data);  
            $this->load->view('pages/notificar');  
            $this->load->view('templates/footer');  
        } else {  
            redirect('error403');  
        }  
    }  
  
    public function constancias()  
    {  
        if ($this->session-  
>userdata('ci_session_key_generate') == false) {  
            redirect('ingresar');
```



```
        } elseif ($this->session-  
>userdata('ci_seesion_key')['tipo'] == 1 || $this->session-  
>userdata('ci_seesion_key')['tipo'] == 3) {  
            $data['titulo'] = "Constancia";  
            $this->load->view('templates/header', $data);  
            $this->load->view('pages/constancia');  
            $this->load->view('templates/footer');  
        } else {  
            redirect('error403');  
        }  
    }  
}
```

## MODELO

```
// create new Obra  
public function createObra()  
{  
    $data = array(  
        'nom_obra' => $this->_nomobra,  
        'mes_obra' => $this->_mesobra,  
        'anio_obra' => $this->_anioobra,  
        'obs_obra' => $this->_obsobra,  
    );  
    $this->db->insert('obra', $data);  
    return $this->db->insert_id();  
}  
// Modificar Obra  
public function updateObra()  
{  
    $data = array(  
        'nom_obra' => $this->_nomobra,  
        'mes_obra' => $this->_mesobra,  
        'anio_obra' => $this->_anioobra,  
        'obs_obra' => $this->_obsobra,  
    );  
    $this->db->where('cod_obra', $this->_codobra);  
    $this->db->update('obra', $data);  
}  
// Mostrar Obra  
public function getObra()  
{  
    $this->db->  
>select(array('o.cod_obra as codobra', 'o.nom_obra as nomobra', 'o.mes_obra as mesobra', 'o.anio_obra as anioobra', 'o.obs_obra as obsobra'));  
    $this->db->from('obra o');
```



```
        $this->db->where('o.cod_obra', $this->_codobra);
        $query = $this->db->get();
        return $query->row_array();
    }

    // Eliminar Obra
    public function deleteObra()
    {
        $this->db->where('cod_obra', $this->_codobra);
        $this->db->delete('obra');
    }

    // crear nuevo Propietario
    public function crearpropietario()
    {
        $data1 = array(
            'tipo_dir' => $this->_tipo,
            'nom_dir' => $this->_nombrevia,
            'nro_dir' => $this->_numero,
            'marg_dir' => $this->_margen,
            'cuad_dir' => $this->_cuadra,
            'obs_dir' => $this->_observacion,
        );
        $this->db->insert('direccion', $data1);
        $this->_fcoddir = $this->db->insert_id();
        $data2 = array(
            'dni_prop' => $this->_codigo,
            'data_prop' => $this->_apellidosynombres,
            /* 'nom_prop' => $this->_nombres,
            'apm_prop' => $this->_apellidos, */
            'dir_prop' => $this->_fcoddir,
            'long_prop' => $this->_longitud,
            'avia_prop' => $this->_via,
            'aver_prop' => $this->_vereda,
        );
        $this->db->insert('propietario', $data2);
        return $this->db->insert_id();
    }

    // Modificar Propietario
    public function actualizarpropietario($coddir)
    {
        $data1 = array(
            'tipo_dir' => $this->_tipo,
            'nom_dir' => $this->_nombrevia,
            'nro_dir' => $this->_numero,
            'marg_dir' => $this->_margen,
            'cuad_dir' => $this->_cuadra,
            'obs_dir' => $this->_observacion,
        );
    }
```





```
$this->db->where('cod_dir', $coddir);
$this->db->update('direccion', $data1);
$data2 = array(
    'dni_prop' => $this->_codigo,
    'data_prop' => $this->_apellidosynombres,
    //'apm_prop' => $this->_apellidos,
    'long_prop' => $this->_longitud,
    'avia_prop' => $this->_via,
    'aver_prop' => $this->_vereda,
);
$this->db->where('dir_prop', $coddir);
$this->db->update('propietario', $data2);
}
// ver Propietario
public function verprop()
{
    $this->db->select(
        array(
            'p.dni_prop dni',
            'p.data_prop apellidosynombres',
            //'p.apm_prop apellidos',
            'd.tipo_dir tipo',
            'd.nom_dir nombrevia',
            'd.nro_dir nrodireccion',
            'd.marg_dir margen',
            'd.cuad_dir cuadra',
            'd.obs_dir observacion',
            'p.long_prop longitud',
            'p.avia_prop via',
            'p.aver_prop vereda',
            'd.cod_dir coddir',
        )
    );

    $this->db->from('propietario as p');
    $this->db->join('direccion d', 'p.dir_prop = d.cod_dir');
    $this->db->where('p.dni_prop', $this->_codigo);
    $query = $this->db->get();
    return $query->row_array();
}
// Eliminar Propietario
public function eliminarprop()
{
    $this->db->where('cod_dir', $this->_codigo);
    $this->db->delete('direccion');
}
#buscar Propietario
public function buscar_prop($apellidospr)
{
```



```
$this->db->like('data_prop', $apellidospr, 'both');
$this->db->order_by('data_prop', 'ASC');
$this->db->limit(10);
$this->db-
>join('direccion', 'direccion.cod_dir = propietario.dir_prop');
return $this->db->get('propietario')->result();

/* $busq_approp = $this -> db -
>query("SELECT propietario.*, direccion.* FROM propietario propietario,
direccion direccion WHERE propietario.dir_prop = direccion.cod_dir
and propietario.data_prop LIKE '%" . $apellidospr . "%' ESCAPE '!' LIMIT 1
0");

return $busq_approp->result(); */
}
#Buscar Obra
public function buscar_obra($obra)
{
    $this->db->like('nom_obra', $obra, 'both');
    $this->db->order_by('nom_obra', 'ASC');
    $this->db->limit(10);
    return $this->db->get('obra')->result();
}
#guardar Aporte
public function guardarAporte()
{
    $data = array(
        'obra_apte' => $this->_codobra,
        'prop_apte' => $this->_codprop,
        'via_apte' => $this->_viaapte,
        'ver_apte' => $this->_verapte,
        'tot_apte' => $this->_totapte,
    );
    $this->db->insert('aporte', $data);
    $this->_fcodapte = $this->db->insert_id();
    $data2 = array(
        'apte_dda' => $this->_fcodapte,
        'cuota_dda' => round(($this->_totapte * 0.4), 1),
        'saldo_dda' => $this->_totapte,
    );
    $this->db->insert('deuda', $data2);
    return $this->db->insert_id();
}

#Guardar Pago
public function guardarPago()
{
    $data = array(
        'deuda_pago' => $this->_deudapago,
```



```
'monto_pago' => $this->_montopago,  
'fecha_pago' => $this->_fechapago,  
'comp_pago' => $this->_nrocepago,  
'doc_pago' => $this->_docdipago,  
);  
$this->db->insert('pagos', $data);  
  
$data2 = array(  
    'saldo_dda' => $this->_updatedda,  
);  
$this->db->where('cod_dda', $this->_deudapago);  
$this->db->update('deuda', $data2);  
return $this->db->insert_id();  
}  
#Buscar Comprobante  
public function buscarcomprobante($ctepago)  
{  
    return $this->db->  
>get_where('pagos', array('comp_pago' => $ctepago))->result();  
}  
  
#Buscar deuda  
public function buscar_deuda($apellidosrecibo)  
{  
    $busquedarec = $this->db->  
>query("SELECT obra.*, aporte.*, propietario.*, direccion.*, deuda.* FR  
OM obra obra, aporte aporte, propietario propietario, direccion direcci  
on, deuda deuda WHERE aporte.obra_apte = obra.cod_obra AND aporte  
.prop_apte = propietario.dni_prop AND propietario.dir_prop = direcc  
ion.cod_dir AND deuda.apte_dda = aporte.cod_apte AND propietario.dat  
a_prop LIKE '%" . $apellidosrecibo . "%' ESCAPE '!' LIMIT 10");  
    return $busquedarec->result();  
}  
  
#buscar aporte  
public function buscar_aporte($dni)  
{  
return $this->db->get_where('aporte', array('prop_apte' => $dni))-  
>result();  
;  
}
```

## VISTA



Figura 1. Interfaz de Registrar Obras

## Anexo G: Resultado de la implementación del sistema de información

### Iteración 1

Implementación – Historia de Usuario Numero 1

*Autenticación de Usuario*

Se muestra la interfaz ya implementada de autenticación de Usuario donde le operador del sistema debe ingresar sus credenciales para ingresar al sistema SIRPACEPV 1.0



**Figura 2.** Interfaz de Autenticación de Usuario

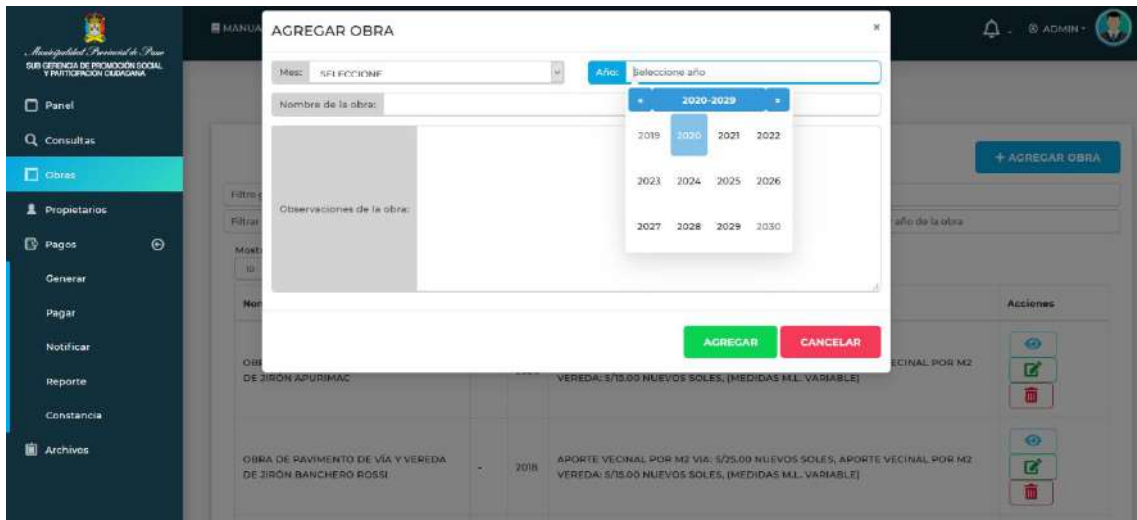
Fuente: Elaboración Propia

### Iteración 2

Implementación de Historia de Usuario 2

*Registrar Obras*

Esta ventana permite registrar las obras con todos los atributos especificados en el requerimiento.



**Figura 3.** Interfaz de Registrar Obras

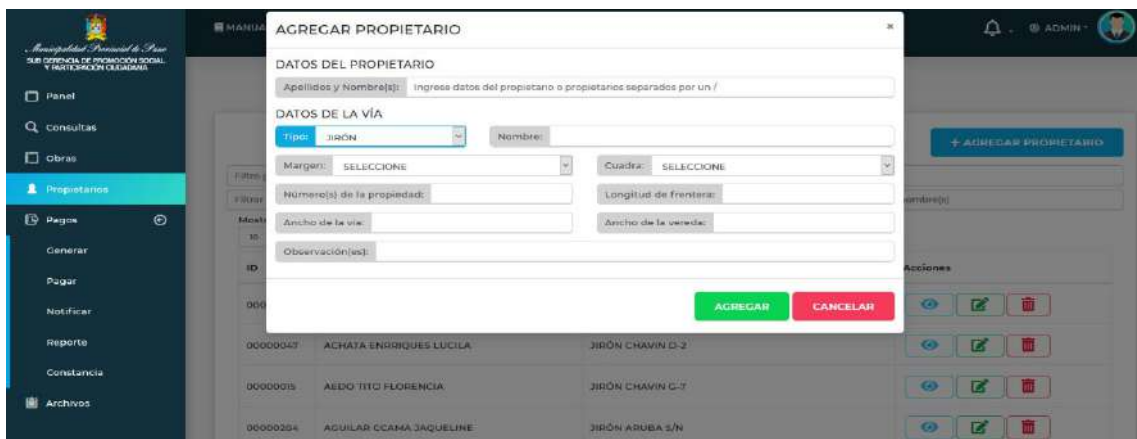
Fuente: Elaboración Propia

## Iteración 2

### Implementación - Historia de Usuario 3

### *Registrar Contribuyente*

Esta ventana de registro de contribuyentes está habilitada para el operador del sistema, que le permite almacenar toda la información relevante del contribuyente.



**Figura 4.** Interfaz para Registrar Contribuyente

Fuente: Elaboración Propia



### Iteración 3

Historia de Usuario 5

*Generar monto de Contribución*

Esta opción está habilitada solo para el operador del sistema que le permite generar el monto de Contribución.

**Figura 5.** Interfaz para Generar monto de Contribución

Fuente: Elaboración Propia

### Iteración 4

Historia de Usuario 6

*Registrar Pagos*

Se presenta la interfaz para registrar los pagos y su comprobante de los contribuyentes.

Figura 6. Interfaz para registrar Pagos

Fuente: Elaboración Propia

#### Iteración 4

Historia de Usuario 6

*Consultar Pagos.*

Esta venta permite al Operador realizar consultas de los pagos de cada contribuyente de manera detallada.

Figura 7. Interfaz para consultar pagos de contribuyentes

Fuente: Elaboración Propia



## Iteración 4

Historia de Usuario 6 *Generar Constancias de cancelación y no adeudo.*

Se muestra la ventana donde Operador podrá emitir constancias de cancelación y no adeudo a los contribuyentes que ya no tengan deuda.

The image shows two screenshots of a web application interface. The top screenshot is titled 'CONSTANCIA DE NO ADEUDO' and the bottom one is 'CONSTANCIA DE CANCELACIÓN'. Both forms have a sidebar on the left with a menu containing 'Panel', 'Consultas', 'Obras', 'Propietarios', 'Pagos', 'Generar', 'Pagar', 'Notificar', 'Reporte', 'Constancia', and 'Archivos'. The 'Pagos' and 'Constancia' items are highlighted in blue. The forms themselves contain several input fields: 'Apellidos del Propietario', 'Deuda' (with a 'S/' indicator), 'Monto recaudado' (with a 'S/' indicator), 'Saldo por recaudar' (with a 'S/' indicator), 'Nro. D.N.I.' (with a 'Nro.' indicator), and 'Número de Registro de Trámite' (with a 'Nro.' indicator). The 'CONSTANCIA DE CANCELACIÓN' form also has a 'Monto a Letras (Opcional)' field. At the bottom of each form are two buttons: 'REINICIAR' (with a circular arrow icon) and 'IMPRIMIR' (with a printer icon).

**Figura 8.** Interfaz para Generar Constancias de cancelación y no adeudo

Fuente: Elaboración Propia

## Iteración 4

Historia de Usuario 6 *Generar Notificaciones.*

Es el resultado del módulo que permite al operador generar notificaciones para los contribuyentes que mantienen deuda.

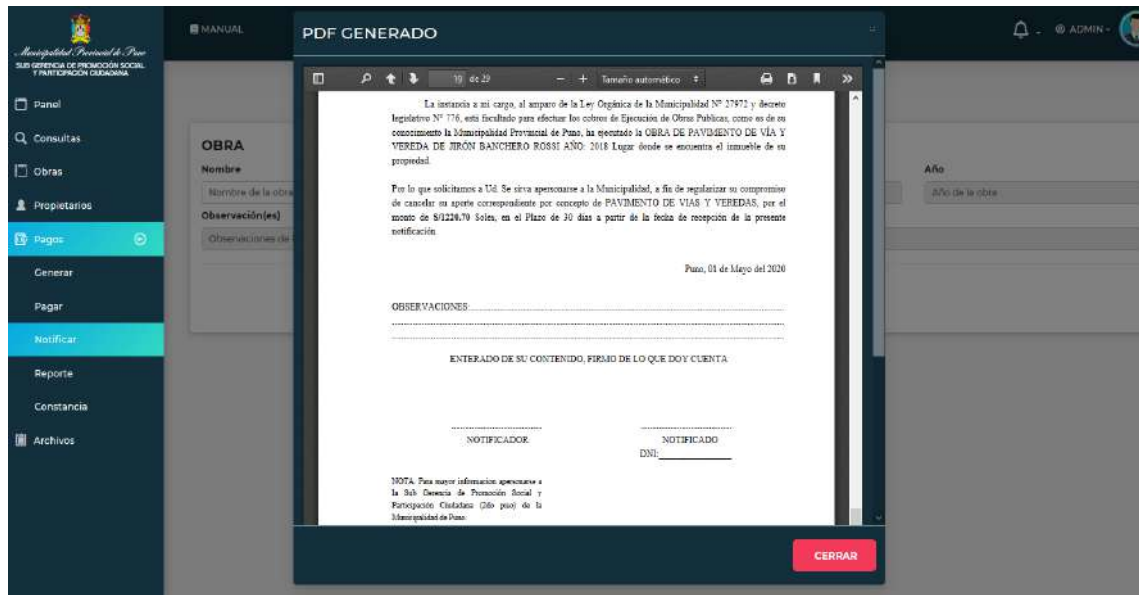


Figura 9. Interfaz para Generar Notificaciones

Fuente: Elaboración Propia

## Iteración 4

Historia de Usuario 6

Generar Reporte de pagos.



Figura 10. Interfaz para Generar Reporte de pagos.

Fuente: Elaboración Propia

## Anexo H: Fichas de evaluación de ISO – 9126.

### Evaluación De Calidad De Software Estándar ISO – 9126

INDICADORES	PREGUNTAS	PUNTUACIÓN				
		1	2	3	4	5
<b>FUNCIONALIDAD</b>						
Aplicabilidad	¿Tiene el conjunto de funciones apropiadas para las tareas específicas?					
Precisión	¿Hace lo que fue acordado en forma esperada y correcta?					
Interoperabilidad	¿interactúa con otros sistemas especificados sin afectar su funcionamiento?					
Seguridad de acceso	¿El sistema puede impedir el acceso a personal no autorizado?					
<b>CONFIABILIDAD</b>						
Madurez	¿El sistema funciona sin presentar fallas?					
Tolerancia de errores	¿Las fallas ocasionales del sistema no afectan gravemente el funcionamiento del sistema?					
Recuperabilidad	¿Es capaz de recuperar daos en caso de fallas?					
<b>USABILIDAD</b>						
Entendimiento	¿Es fácil de entender y reconocer la estructura y la lógica y su aplicabilidad?					
Aprendizaje	¿Es fácil de aprender a usar el sistema?					
Operabilidad	¿Es fácil de operar y controlar el sistema?					
Atracción	¿Es atractivo el diseño del software?					
<b>EFICIENCIA</b>						
Comportamiento de tiempos	¿Qué tan rápido responde el sistema?					
Utilización de recursos	¿Cuántos recursos usa y durante cuánto tiempo?					
<b>MANTENIMIENTO</b>						
Capacidad de ser analizado	¿Es fácil de diagnosticar una falla o identificar partes a modificar?					
Cambiabilidad	¿Es fácil de modificar y adaptar?					
Estabilidad	¿Hay riesgos o efectos inesperados cuando se realizan cambios?					
facilidad de prueba	¿Son fáciles de validar las modificaciones?					
<b>PORTABILIDAD</b>						
Adaptabilidad	¿Es fácil de adaptar a otros entornos con lo provisto?					
facilidad de instalación	¿Es fácil de instalar en el ambiente especificado?					
Reemplazabilidad	¿Es fácil de usarlo en lugar de otro software para ese ambiente?					
Coexistencia	¿El software puede funcionar con otros sistemas?					
Eficiencia	¿La eficaz el software cuando el usuario final realiza los procesos?					



### Anexo I: Datos del tiempo (en minutos) de Demora en la Atención Antes(y) y

Después (x) de la implementación del sistema.

<b>N° DE MUESTRAS</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>	<b>11</b>	<b>12</b>	<b>13</b>	<b>14</b>
<b>ANTES (y)</b>	12	19	20	13	12	11	20	19	20	14	15	13	17	16
<b>DESPUES (x)</b>	4	5	4	5	4	3	3	6	3	6	6	5	4	5
<b>DIFERENCIA</b>	-8	-14	-16	-8	-8	-8	-17	-13	-17	-8	-9	-8	-13	-11

<b>N° DE MUESTRAS</b>	<b>15</b>	<b>16</b>	<b>17</b>	<b>18</b>	<b>19</b>	<b>20</b>	<b>21</b>	<b>22</b>	<b>23</b>	<b>24</b>	<b>25</b>	<b>26</b>	<b>27</b>	<b>28</b>
<b>ANTES (y)</b>	11	17	15	15	12	18	15	13	11	13	17	16	15	13
<b>DESPUES (x)</b>	6	6	5	4	5	4	5	5	3	5	5	5	4	5
<b>DIFERENCIA</b>	-5	-11	-10	-11	-7	-14	-10	-8	-8	-8	-12	-11	-11	-8

<b>N° DE MUESTRAS</b>	<b>29</b>	<b>30</b>	<b>31</b>	<b>32</b>	<b>33</b>	<b>34</b>	<b>35</b>	<b>36</b>	<b>37</b>	<b>38</b>	<b>39</b>	<b>40</b>	<b>41</b>	<b>42</b>
<b>ANTES (y)</b>	10	12	13	18	19	13	17	17	18	12	13	13	20	16
<b>DESPUES (x)</b>	3	4	3	3	5	3	3	3	3	5	5	6	3	4
<b>DIFERENCIA</b>	-7	-8	-10	-15	-14	-10	-14	-14	-15	-7	-8	-7	-17	-12

<b>N° DE MUESTRAS</b>	<b>43</b>	<b>44</b>	<b>45</b>	<b>46</b>	<b>47</b>	<b>48</b>	<b>49</b>	<b>50</b>	<b>51</b>	<b>52</b>	<b>53</b>	<b>54</b>	<b>55</b>	<b>56</b>
<b>ANTES (y)</b>	13	18	12	17	16	13	15	16	17	20	12	14	13	12
<b>DESPUES (x)</b>	5	6	7	3	3	5	4	5	6	6	4	5	6	4
<b>DIFERENCIA</b>	-8	-12	-5	-14	-13	-8	-11	-11	-11	-14	-8	-9	-7	-8

<b>N° DE MUESTRAS</b>	<b>57</b>	<b>58</b>	<b>59</b>	<b>60</b>	<b>61</b>	<b>62</b>	<b>63</b>	<b>64</b>	<b>65</b>	<b>66</b>	<b>PROMEDIO</b>
<b>ANTES (y)</b>	20	15	15	18	19	13	16	20	14	14	15
<b>DESPUES (x)</b>	5	3	4	4	3	3	5	4	5	5	4
<b>DIFERENCIA</b>	-15	-12	-11	-14	-16	-10	-11	-16	-9	-9	-10.78

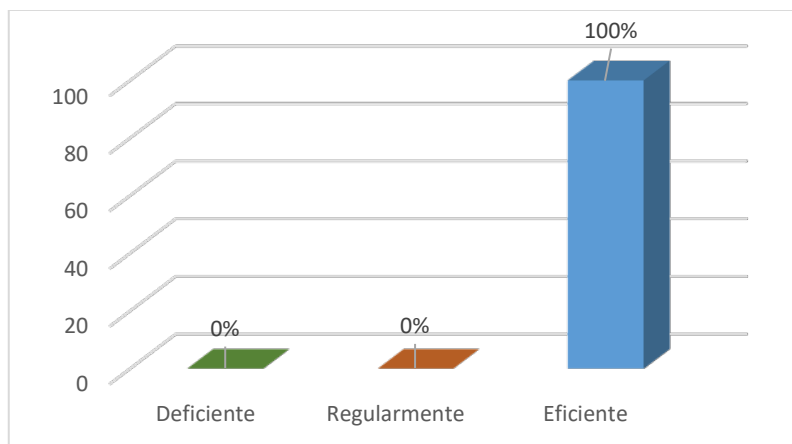
Fuente: Elaboración Propia

**Anexo J: Resultados del cuestionario acerca de “SIRPACEPV”.**

**Tabla 19.** Servicios que ofrece el sistema “SIRPACEPV”

Servicios que ofrece el Sistema	Frecuencia	Porcentaje (%)
Deficiente	0	0
Regularmente	0	0
Eficiente	5	100
<b>Total</b>	<b>5</b>	<b>100</b>

Fuente: Elaboración Propia



**Figura 11.** Servicios que ofrece el sistema "SIRPACEPV"

Fuente: Elaboración Propia

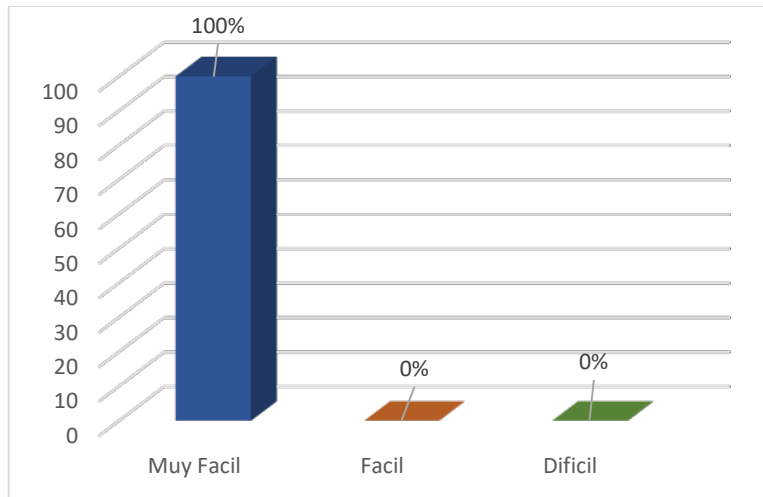
Se puede observar que el 100% de los trabajadores que labora en la sub gerencia de promoción social y participación ciudadana de la Municipalidad Provincial De Puno, considera que el servicio que ofrece el sistema “SIRPACEPV” es eficiente.

**Tabla 20.**

Interacción con el sistema.

Interacción con el Sistema	Frecuencia	Porcentaje (%)
Muy Fácil	5	100
Fácil	0	0
Difícil	5	0
<b>Total</b>	<b>5</b>	<b>100</b>

Fuente: Elaboración propia



**Figura 12.** Interacción con el sistema "SIRPACEPV"

Fuente: Elaboración Propia

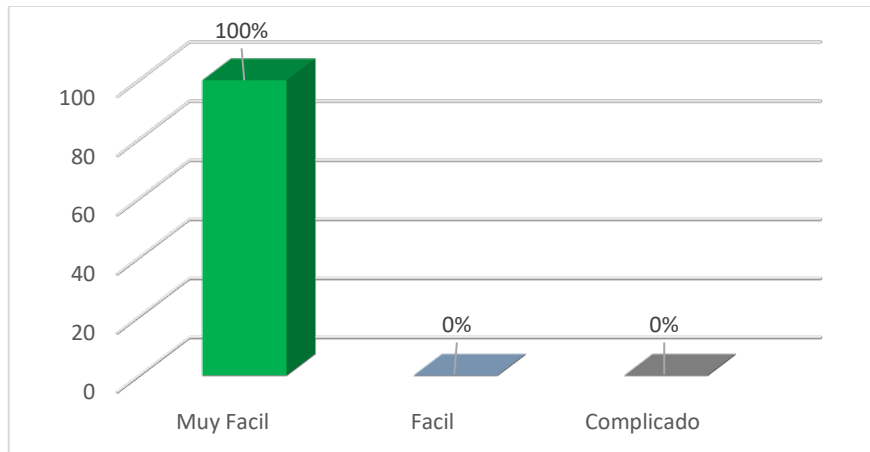
De acuerdo con la encuesta realizada en la oficina de sub gerencia de promoción social y participación ciudadana de la Municipalidad Provincial de Puno, se puede observar que el 100% de los Operadores considera que la interacción con el sistema “SIRPACEPV” es fácil.

**Tabla 21.**

Ingreso de los datos y/o registros al sistema.

Ingreso de datos	Frecuencia	Porcentaje (%)
Muy Fácil	5	100
Fácil	0	0
Complicado	5	0
<b>Total</b>	<b>5</b>	<b>100</b>

Fuente: Elaboración Propia



**Figura 13.** Ingreso de los datos y/o registros al sistema  
Fuente: Elaboración Propia

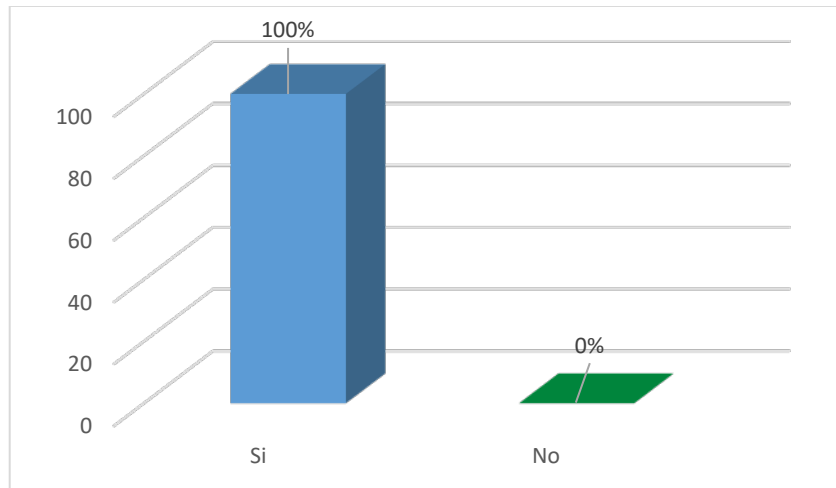
Podemos ver que, según los encuestados que laboran en la subgerencia de promoción social y participación ciudadana de la Municipalidad Provincial de Puno, el 100% considera que el ingreso de datos y/o registro que ofrece el sistema “SIRPACEPV” es Muy Fácil.

**Tabla 22.**

Confiabilidad de registro de pagos en el sistema “SIRPACEPV”.

Confiabilidad de pagos en Sistema	Frecuencia	Porcentaje (%)
Si	5	100
No	0	0
<b>Total</b>	<b>5</b>	<b>0</b>

Fuente: Elaboración Propia



**Figura 14.** Confiabilidad en el registro de pagos en sistema "SIRPACEPV"  
Fuente: Elaboración Propia

De acuerdo a la encuesta aplicada al personal que labora en la subgerencia de promoción social y participación ciudadana de la Municipalidad Provincial de Puno, se observa que el 100% de los trabajadores considera que el registro de pagos que ofrece el sistema "SIRPACEPV" si son confiables.

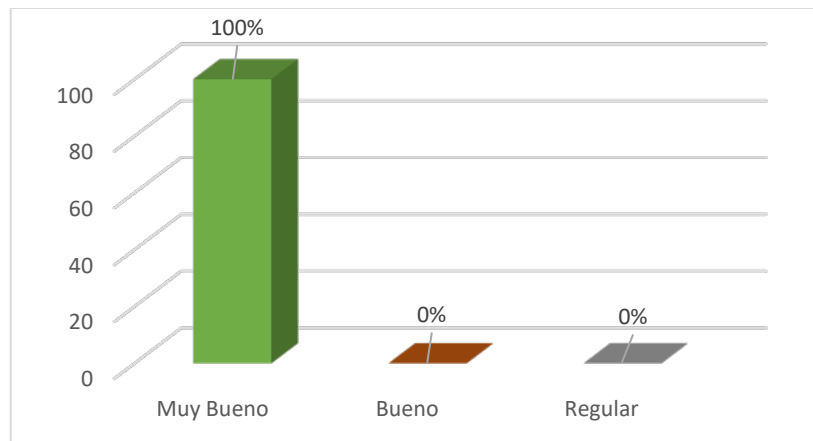
**Tabla 23.**

Reportes que emite el sistema "SIRPACEPV".

Reportes satisfactorios	Frecuencia	Porcentaje (%)
Muy Bueno	5	100
Bueno	0	0
Regular	5	0
<b>Total</b>	<b>5</b>	<b>100</b>

Fuente: Elaboración propia





**Figura 15.** Reportes que emite el sistema "SIRPACEPV"

Fuente: Elaboración propia

Se puede ver que, según los encuestados que laboran en la subgerencia de promoción social y participación ciudadana de la Municipalidad Provincial de Puno, el 100% considera que los reportes que emite el sistema "SIRPACEPV" es Muy Bueno.



## Anexo K: Manual de Usuario “SIRPACEPV”.

### INTRODUCCIÓN

El presente manual muestra los pasos a seguir para realizar todas las tareas a nivel administrador dentro el sistema SIRPACEPV. Esto con la finalidad de brindar al usuario una herramienta que asegure el uso correcto del sistema.

### OBJETIVO GENERAL

Este manual tiene como objetivo proporcionar al usuario una herramienta que facilite el ingreso de información en cada una de las opciones que conforman el Sistema SIRPACEPV.

### ALCANCE

Este manual va dirigido a usuarios a los usuarios que interactuaran con el sistema, y para agilizar el aprendizaje en el ingreso de información al sistema.

Este manual contiene las instrucciones para utilizar correcta y adecuadamente el sistema SIRPACEPV, introduciendo al usuario en la metodología de ingreso y validación de información requerida por el programa informático, así como las diferentes opciones y utilidades que el Sistema Proporciona.

### DEFINICIÓN DE TÉRMINOS

**Sistema:** Almacena y procesa información; es el conjunto de partes interrelacionadas en este caso hardware, software y recurso humano.

**SIRPACEPV:** Sistema de registro de pagos de contribución especial de pistas y veredas.

## CAPITULO I - GENERALES DEL SISTEMA

### 1.1 Requisitos mínimos del Sistema

Los requerimientos mínimos para que el sistema funcione correctamente, son los siguientes:

- 1) Computadora con conexión a internet.
- 2) Algún Navegador web (Se recomienda Google Chrome Última versión).
- 3) Cuenta de Usuario.

### 1,2 ¿A quién está dirigido este Manual?

Este manual está dirigido a las personas responsables de hacer el registro de información de pagos por contribución especial de pistas y veredas en el sistema SIRPACEPV.

### 1.3 Acceso y Conectividad al Sistema

Antes de ingresar al Sistema SIRPACEPV, es importante Ingresar la siguiente dirección en el navegador <http://192.168.100.16/>

## CAPITULO II - COMO REGISTRAR LA INFORMACIÓN EN EL SISTEMA (SIRPACEPV)

### 2.1 ¿Cómo se ingresa al Sistema?

Antes de ingresar al Sistema SIRPACEPV, es importante ingresar la siguiente dirección en el navegador <http://192.168.100.16/> aparecerá una pantalla.



#### Paso1. (Ingreso al Sistema)



Escriba en la pantalla de inicio de sesión el Usuario y Contraseña, luego haga clic sobre el botón "Ingresar".

Municipalidad Provincial de Puno  
SUB GERENCIA DE PROMOCIÓN SOCIAL  
Y PARTICIPACIÓN CIUDADANA

Usuario y/o contraseña incorrectos

USUARIO  
Ingrese nombre de usuario.

CONTRASEÑA  
Ingrese contraseña.

INGRESAR

Información del sistema  
Versión: 1.0.0  
Dirección IP: 179.7.224.213  
Hora: 16:30:04  
Fecha: 12/12/2019

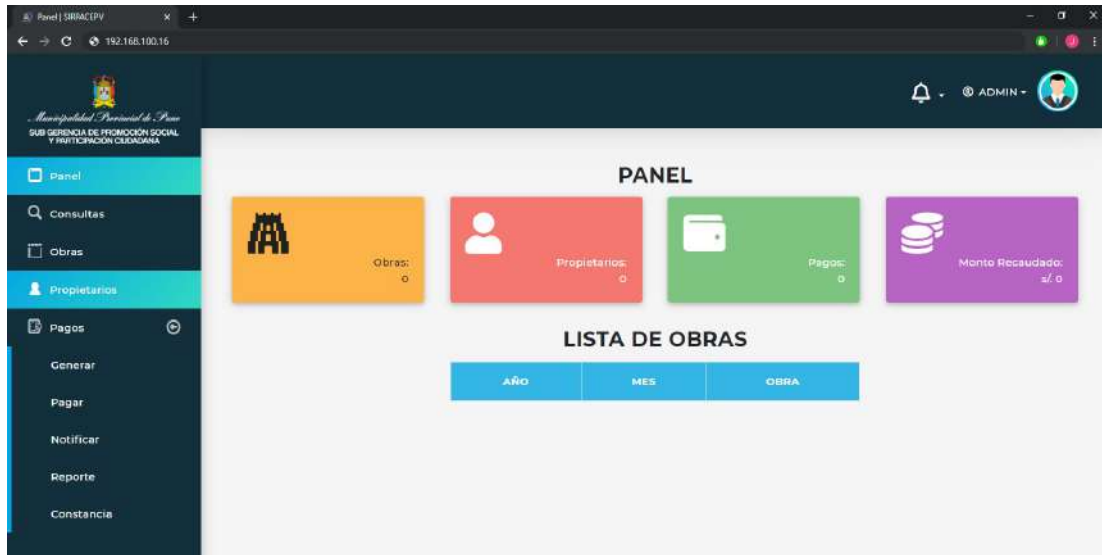
© SIRPACEPV  
PUNO - PERÚ



Introducir el nombre de usuario, contraseña y presionar el botón “Ingresar”. El sistema comprobará si los datos introducidos coinciden si existe en la base de datos. Si en caso hubiera error, el sistema lo hará saber con un mensaje como el que se muestra en la fig.3. En ese caso deberá escribir de nuevo los datos.

## Paso 2. (Ingreso al Sistema)

Si los datos fueron correctos, el sistema redireccionará al apartado donde muestra el panel principal.



## 2.2 Ingreso de información de Obras

- 1) Para registrar una obra nos dirigimos al botón Obras
- 2) Para agregar una obra hacemos click en el botón “+ AGREGAR OBRA”.
- 3) Una vez llenado toda la información necesaria hacemos click en el botón “AGREGAR”.

- 4) Obtendrá un mensaje de alerta indicando que se registró correctamente la obra.
- 5) En seguida podrá observar la obra que agrego con sus tres acciones (ver, editar, eliminar).

AGREGAR PROPIETARIO

DATOS DEL PROPIETARIO

Número de DNI: 73658877

Apellidos del propietario: Zapana Mamani

Nombre(s) del propietario: Jose Luis

DATOS DE LA VÍA

Tipo: AVENIDA Nombre: AV. CANCHARANI

Margen: DERECHO Cuadra: PRIMERA

Número(s) de la propiedad: 539 Longitud de frentera: 8

Ancho de la vía: 4 Ancho de la vereda: 2

Observacion(es):

AGREGAR CANCELAR

- 1) Para registrar a un propietario nos dirigimos a botón propietarios.
- 2) Para agregar la información de un propietario hacemos click en “+ AGREGAR PROPIETARIO”.
- 3) Una vez llenado toda la información necesaria hacemos click en el botón “AGREGAR”.

MÓDULO PROPIETARIOS

+ AGREGAR PROPIETARIO

Filtro general

Filtrar por nro. DNI Filtrar por apellido Filtrar por nombre(s)

Mostrar: ID

DNI	Apellidos y Nombre(s)	Dirección	Acciones
73658877	ZAPANA MAMANI, JOSE LUIS	AVENIDA AV. CANCHARANI 539 MARGEN DERECHO PRIMERA CUADRA	
DNI	Apellidos y Nombre(s)	Dirección	Acciones

Mostrando registros del 1 al 1 de un total de 1 registros

Anterior 1 Siguiente

- 4) Obtendrá un mensaje de alerta indicando que se registró correctamente el propietario.
- 5) En seguida podrá observar al propietario que agrego con sus tres acciones (ver, editar, eliminar).





## 2.5 Registro de pagos

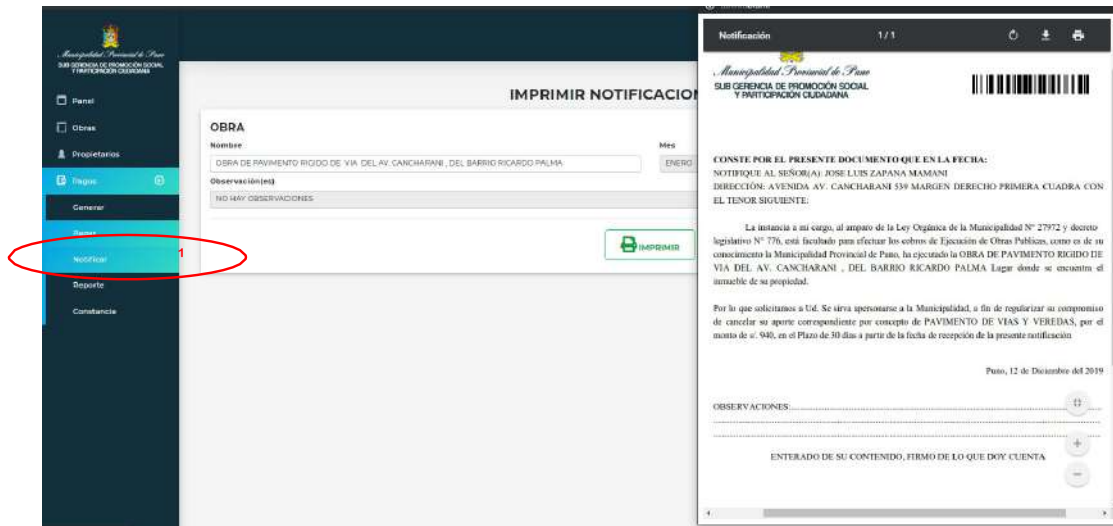
The screenshot shows a web interface for recording payments. The form is titled 'REGISTRAR PAGO' and is divided into several sections. A sidebar on the left contains navigation options: 'Panel', 'Obras', 'Propietario', 'Pagos', 'Generar', 'Pagar', 'Notificar', 'Reporte', and 'Constancia'. The 'Pagos' option is highlighted with a red circle and the number 1. The main form area is divided into three main sections: 'DATOS DE LA DEUDA - APOORTE GENERADO', 'DATOS DEL PAGO', and 'COMPROBANTE DIGITALIZADO'. The 'DATOS DE LA DEUDA - APOORTE GENERADO' section contains fields for 'Número de D.N.I.' (with value 73658277 and a red circle 2), 'Cuota 40% del costo total' (with value S/ 436.00), 'Costo total por la obra' (with value S/ 1040.00), and 'Monto recaudado' (with value S/ 0). The 'DATOS DEL PAGO' section contains fields for 'Monto pagado' (with value S/ 100 and a red circle 3), 'Fecha de pago' (with value 12/10/2019 and a red circle 4), and 'Comprobante de pago' (with value 326456789). The 'COMPROBANTE DIGITALIZADO' section contains a scanned receipt image (with a red circle 6) and a green confirmation message 'Correcto Se guardo el pago'. At the bottom of the form, there are two buttons: 'REINICIAR' and 'GUARDAR PAGO' (with a red circle 7).

- 1) Para registrar el pago realizado por el propietario nos dirigimos al botón pagos y pagar.
- 2) Ingresamos el número de DNI del propietario para que el sistema nos proporcione sus datos.
- 3) Ingresar la cantidad pagada por el propietario.
- 4) Ingresar La fecha de pago según el comprobante de pago.
- 5) Ingresar el número de comprobante de pago.
- 6) Subimos el comprobante escaneado.
- 7) Para guardar el pago hacemos click en el botón “GURADAR PAGO”, y el sistema le mostrará un mensaje validando su acción.



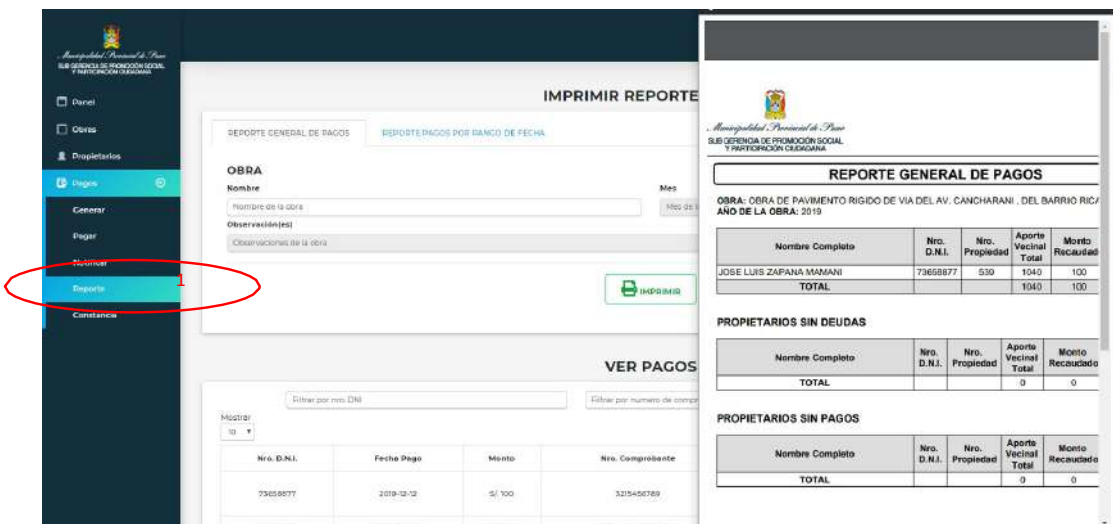
## CAPITULO III - COMO IMPRIMIR REPORTES Y DOCUMENTOS EN (SIRPACEPV)

### 3.1 Imprimir notificaciones



- 1) Para imprimir notificaciones para los propietarios nos dirigimos al botón pagos y notificar.
- 2) Llamamos la obra y click en “Imprimir”.

### 3.2 Imprimir reporte de pagos



- 1) Para imprimir el reporte de pagos de propietarios nos dirigimos al botón pagos y reporte.
- 2) Llamamos la obra y click en el botón “Imprimir”.

### 3.3 Imprimir constancia de no adeudar



- 1) Para imprimir la constancia de no adeudar ingresamos el DNI del propietario y click en "IMPRIMIR".