

**UNIVERSIDAD NACIONAL DE CHIMBORAZO**



**FACULTAD DE INGENIERIA  
CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN**

“Proyecto de Investigación previo a la obtención de título de Ingeniero en Sistemas y Computación”

**TRABAJO DE TITULACIÓN**

**“MODELO FURPS PARA EVALUAR EL SISTEMA WEB DE RECAUDACION DE PATENTES GADM PENIPE”**

**Autor:**

Alex Bladimir Manobanda Ushca

**Tutor:**

Ing. Pamela Buñay Guisñan.

**Riobamba - Ecuador**

**2020**

## **DERECHO DE AUTORÍA**

La responsabilidad del contenido de este proyecto de graduación corresponde exclusivamente a: Alex Bladimir Manobanda Ushca autor, y la MsC. Pamela Buñay Guisñan, Directora de Tesis, y al patrimonio intelectual de la Universidad Nacional de Chimborazo.

### **Autores**



.....  
Alex Bladimir Manobanda Ushca  
060441383-1

### **Directora del Proyecto:**



.....  
MsC. Pamela Alexandra Buñay Guisñan  
060424673-6

## **DEDICATORIA**

El presente trabajo de investigación dedico en primer lugar a Dios, por darme y la vida y permitirme llegar hasta esta instancia en mi formación académica.

A mis padres José y María quienes son mi pilar fundamental en el día a día y en cada logro de mi vida, por darme la mejor herencia que un hijo puede tener que es la educación, a mi hermano Junior por siempre estar ahí apoyándome y brindándome sus consejos y su apoyo incondicional durante todo este duro sendero, a mi familia que siempre estuvo impulsándome y apoyándome de una u otra manera para completar este sueño.

A mis docentes por compartirme sus conocimientos y impulsarme a ser mejor. A mis compañeros y amigos que fueron parte importante durante esta vida universitaria, por todos los momentos compartidos, por su amistad, su confianza y su ayuda mutua.

**Alex Bladimir Manobanda Ushca**

## **AGRADECIMIENTO**

Agradezco a Dios por cuidarme, guiarme e iluminarme a lo largo de mi vida y mis estudios, agradezco a mi familia por brindarme su apoyo siempre.

A la Universidad Nacional de Chimborazo especialmente a la carrera de Ingeniería en Sistemas y Computación, por abrirme las puertas para forjar mis conocimientos profesionales como también personales.

Un agradecimiento al GADM – Penipe, al departamento de TI por darme la apertura y apoyo para poder realizar este proyecto y culminarlo con éxito.

Agradezco a la Ing. Pamela Buñay por brindarme sus conocimientos, su apoyo, su confianza y guiarme para culminar exitosamente este trabajo.

Un agradecimiento especial a Karina Pinduisaca persona super importante en mi vida, gracias por todo su apoyo durante toda esta vida universitaria.

A mis compañeros de clase con quienes empecé esta epata y con los que he compartido grandes recuerdos gracias por sus consejos y apoyo mutuo.

**Alex Bladimir Manobanda Ushca**

## ÍNDICE GENERAL

DERECHO DE AUTORÍA .....	II
DEDICATORIA .....	III
AGRADECIMIENTO .....	IV
RESUMEN .....	IX
ABSTRACT.....	X
INTRODUCCIÓN .....	1
CAPITULO I .....	2
1. Planteamiento del problema.....	2
1.1 Problema.....	2
1.2 Justificación.....	3
OBJETIVOS .....	4
Objetivo General.....	4
Objetivos Específicos.....	4
CAPITULO II.....	5
2. Marco Teórico.....	5
2.1 Calidad .....	5
2.1.1 Términos relacionados con la Calidad.....	5
2.2 Calidad de Software .....	6
2.3 Atributos de Calidad.....	6
2.4 Modelos de Calidad.....	7
2.5 Modelos de Calidad de Software .....	7
2.6 Modelo FURPS.....	8
2.6.1 Rendimiento Web .....	8
2.6.2 Valores de ponderación del rendimiento según el modelo FURPS.....	9
2.6.3 Ventajas y Desventajas del Modelo FURPS.....	10
2.7 Diseño de Software .....	10
2.7.1 Modelo Cascada.....	10
2.7.2 Arquitectura cliente-servidor.....	12
2.7.3 MVC .....	13
2.8 Desarrollo de Software.....	13
2.8.1 MySQL .....	13
2.8.2 NetBeans .....	14
2.8.3 JSF.....	15
2.9 Aplicaciones para medición del rendimiento.....	16

2.9.1	JMeter .....	16
2.9.2	LoadRunner.....	16
2.9.3	Programas para medición del rendimiento. ....	16
2.10	Metodología para la evaluación del rendimiento con JMeter. ....	16
CAPITULO III.....		18
3	Metodología .....	18
3.1	Tipo y diseño de la investigación.....	18
3.2	Según el objeto de estudio .....	18
3.3	Población de estudio y tamaño de la muestra. ....	18
3.4	Técnicas de recolección de Datos .....	18
3.4.1	Observación. ....	19
3.4.2	Entrevista. ....	19
3.5	Técnica de Análisis e interpretación de la información.....	19
3.5.1	Herramientas Utilizadas.....	19
3.6	Desarrollo de Software .....	19
3.6.1	Análisis de requerimientos.....	20
3.6.2	Diagramación.....	22
3.6.3	Diseño .....	23
3.7	Implementación.....	25
3.8	Pruebas.....	27
3.9	Mantenimiento.....	27
CAPITULO IV.....		28
4	Resultados y Discusión .....	28
4.1	Análisis de los indicadores.....	29
4.1.1	Dimensión eficacia.....	29
4.1.2	Tiempo de Respuesta .....	29
4.1.3	Utilización de Recursos .....	30
CONCLUSIONES .....		32
RECOMENDACIONES.....		33
5	BIBLIOGRAFIA .....	34
ANEXOS .....		36

## ÍNDICE DE TABLAS

Tabla 1. Criterios asociados a factores de calidad FURPS.....	9
Tabla 2: Valores de ponderación establecidos según el modelo FURPS. ....	10
Tabla 3: Comparación de JMeter y LoadRunner. ....	16
Tabla 4: Requisitos Funcionales del sistema. ....	20
Tabla 5: Requisitos No Funcionales del sistema. ....	21
Tabla 6: Casos de Prueba.....	28
Tabla 7: Dimensión Eficacia.....	29
Tabla 8: Utilización de recursos. ....	30
Tabla 9: Comparación de los valores establecidos en el modelo FURPS y los valores obtenidos en la investigación.....	31

## ÍNDICE DE ILUSTRACIONES

Figura 1: Ciclo de vida metodología cascada (Sari, 2017). .....	11
Figura 2: Arquitectura Cliente-Servidor. ....	12
Figura 3: Modelo vista controlador.....	13
Figura 4: MySQL.....	14
Figura 5: Escenario de trabajo en NetBeans. ....	15
Figura 6: Fases modelo cascada.....	19
Figura 7: Flujo de Procedimiento de Patentes .....	20
Figura 8: Caso de uso Sistema de recaudación de patentes. ....	22
Figura 9: Caso de uso Administrador. ....	23
Figura 10: Caso de uso Recaudador.....	22
Figura 11: Vista Conceptual. ....	24
Figura 12: Vista Física. ....	24
Figura 13: Diagrama de clases.....	25
Figura 14: Diagrama de base de datos. ....	26
Figura 15: Resultado de pruebas Iniciales. ....	28
Figura 16: Dimensión Eficacia .....	29
Figura 17: Tiempo de Respuesta.....	30
Figura 18: Reporte Resumen .....	36
Figura 19: Medición Eficacia.....	36
Figura 20: Medición Tiempo de respuesta.....	40
Figura 21: Medición Tiempo de respuesta.....	40
Figura 22: Medición Tiempo de respuesta.....	41
Figura 23: Uso de CPU.....	41
Figura 24: Uso de memoria.....	42
Figura 25: Uso de Disco. ....	42
Figura 26: Certificado de Desarrollo del Sistema Web. ....	43



## RESUMEN

Para los GADS Municipales el desarrollo y la implementación de sistemas web ha pasado de ser una alternativa a ser algo casi esencial, actualmente invertir en tecnología es invertir en eficiencia, lastimosamente el GADM Penipe diariamente realiza una gran cantidad de operaciones de recaudación de patentes, no cuenta con un sistema que permita automatizar este proceso que contribuya a mantener organizada la información, obtener información y reducir tiempo a los contribuyentes.

Por tal motivo el objetivo de la investigación es implementar un sistema web para la recaudación de patentes en el GADM Penipe aplicando la metodología de desarrollo en cascada, la arquitectura de software cliente-servidor y el modelo de calidad de software FURPS para analizar el rendimiento del sistema.

FURPS es un modelo de calidad de software que bajo el acrónimo de su nombre establece un conjunto de factores y atributos de calidad como la Funcionalidad, Usabilidad, Confiabilidad, Rendimiento y Soporte. Para la presente investigación se midió el rendimiento con sus criterios establecidos por el modelo para garantizar la calidad del sistema.

Con la herramienta JMeter se procedió a realizar las pruebas, con una muestra de 10 peticiones cada 5 segundos. Se comprobó que el sistema cumple con el 100% de eficacia, el tiempo de respuesta fue de 993.5 ms y en la utilización de recursos el promedio de uso del CPU, memoria y disco se obtuvo un promedio del 10.6%. Estos resultados permitieron determinar que los porcentajes obtenidos son inferiores a los porcentajes establecidos en el modelo FURPS.

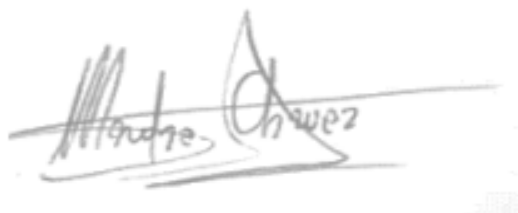
**Palabras Claves:** Calidad de Software, Modelo FURPS, Sistema Web, Rendimiento

## ABSTRACT

For the Municipal GADS the development and implementation of web systems has been an alternative to being something almost essential, currently investing in technology is investing in efficiency, unfortunately the GADM Penipe daily performs a large number of patent collection operations, there is not a system that allows automating this process that helps to keep information organized, obtain information and reduce taxpayers' time. For this reason, the objective of the research was to implement a web system for the collection of patents in the GADM Penipe applying the cascade development methodology, the client-server software architecture and the FURPS software quality model to analyze the performance of the system.

FURPS is a software quality model that under the acronym of its name establishes a set of quality factors and attributes such as Functionality, Usability, Reliability, Performance and Support. For the present investigation, the performance was measured with the criteria established by the model to guarantee the quality of the system. With the JMeter tool, the tests carried out, with a sample of 10 requests every 5 seconds. It found that the system complies with 100% efficiency, the response time was 993.5 ms and in the use of resources the average use of the CPU, memory and disk obtained an average of 10.6%. These results allowed determining that the percentages obtained were lower than the percentages established in the FURPS model.

**Key Words:** Software Quality, FURPS Model, Web System, Performance.

A handwritten signature in black ink, appearing to read "Maritza Chávez". The signature is stylized with a large, looped initial 'M' and a prominent 'C' for 'Chávez'. There are some horizontal lines and scribbles around the signature.

Reviewed by: Chávez, Maritza  
Language Center Teacher

## INTRODUCCIÓN

La era tecnológica en los últimos años ha crecido notablemente, donde los principales beneficiados son las grandes, medianas y pequeñas empresas u organizaciones tanto públicas como privadas, por tal motivo dichos beneficiarios intentan constantemente disponer de herramientas, sistemas que permitan estar adaptados al cambio tecnológico.

Los GADS Municipales son los encargados de satisfacer las necesidades de un cantón, han impulsado la implementación de ciertas herramientas tecnológicas para automatizar sus procesos, que dispondrán de ciertas ventajas, como poder ofrecer una mejor atención a sus mandantes y administrar de mejor manera sus recursos.

La capacidad de automatizar procesos en un contexto organizacional es la razón por la que el desarrollo de sistemas está tomando un papel central en algunos esfuerzos de transformación. Obviamente, la idea de basarse en la estandarización de procesos y modelo de desarrollo, demanda recursos humanos, económicos y tiempo, todos ellos basados en los procesos de control de calidad tanto de productos como de servicios que se deben cumplir para garantizar la calidad del sistema.

FURPS desarrolla un conjunto de factores de calidad software, bajo el acrónimo FURPS, estos factores de calidad, junto a sus atributos son usados para establecer métricas de calidad en las actividades del proceso del desarrollo del software. (Constanzo, 2014)

El presente proyecto de investigación tiene como objetivo automatizar la recaudación de patentes en el GADM Penipe puesto que los procesos se realizan de manera manual desaprovechando tiempo y recursos. Por tal motivo se plantea el desarrollo un sistema web para la recaudación de patentes municipales y evaluar el rendimiento mediante el modelo de calidad de software FURPS.

El presente trabajo de Investigación se estructura de la siguiente manera: Resumen, Introducción, el capítulo I que contiene el planteamiento del problema y los objetivos de la investigación, el capítulo II engloba todo el marco teórico, el capítulo III comprende la metodología empleada en el desarrollo del proyecto, el capítulo IV trata sobre los resultados y discusión de la investigación y por último las conclusiones, recomendaciones y anexos.

## **CAPITULO I**

### **1. Planteamiento del problema**

#### **1.1 Problema**

El desarrollo y la implementación de los sistemas web en la actualidad se han convertido en la base tecnológica de los GAD Municipales. Desarrollar estas plataformas digitales es invertir en eficiencia, las mismas mejoran los procesos y alcances funcionales de los GAD Municipales. (Garita-Araya, 2013)

Los GAD Municipales apuntan cada vez más en adoptar un sistema que automatice sus procesos y mejore los servicios, el desarrollo de una aplicación web o de escritorio está dejando de ser una alternativa para pasar a ser un requerimiento casi esencial. El rendimiento de los sistemas web es de gran importancia tanto para los administradores como para los usuarios, garantizando el ahorro de tiempo en respuesta a las solicitudes y una mayor disponibilidad, haciendo que la más beneficiada sea la Institución. (Batallas, 2013)

Lastimosamente en el GADM Penipe que diariamente realiza una gran cantidad de operaciones de recaudación de patentes no cuenta con un sistema que automatice este proceso, por tal motivo, las operaciones son realizadas manualmente provocando que la información no esté organizada y no sea íntegra, aumentando los tiempos de respuesta y pérdida de documentos, ocasionado que los procesos se vuelvan más lentos y costosos tanto para los funcionarios del GADM Penipe y los ciudadanos. Como propuesta al problema la investigación se orienta al desarrollo de un sistema web de recaudación de patentes que facilitará el trabajo a los funcionarios y brindará un mejor servicio a los ciudadanos del cantón obteniendo como beneficio la disponibilidad de la información íntegra, disminución de errores, agilidad en los procesos y la ejecución de tareas diarias con calidad. Una vez culminada la etapa de desarrollo del sistema se evaluó el rendimiento mediante la aplicación del modelo de calidad de software FURPS.

## **1.2 Justificación**

La ejecución de procesos en el GAD Municipal no responde de forma adecuada a las necesidades que se presentan según su actividad, de manera que gran parte de los procesos se realizan de forma manual, por lo tanto, es importante que se cuente con herramientas que permitan automatizar tareas y procesos que consumen mucho tiempo y recurso del GAD Municipal.

Con la automatización de estos procesos que manualmente tardan mucho tiempo en ejecutarse, es posible ahorrar recursos para la ejecución de otras actividades dentro del GAD Municipal.

Una vez realizado este análisis se desarrolló un sistema web para la recaudación de patentes en el GAD Penipe tomando en cuenta el rendimiento del software según los criterios establecidos en el modelo de calidad de software FURPS como son: eficacia, tiempo de respuesta y utilización de recursos.

## **OBJETIVOS**

### **Objetivo General**

Analizar el Modelo FURPS para evaluar el sistema web de Recaudación de Patentes GADM Penipe.

### **Objetivos Específicos**

- Analizar los factores de calidad del modelo FURPS.
- Desarrollar el sistema web para la Recaudación de Patentes GADM Penipe.
- Medir el rendimiento del software para la recaudación de patentes aplicando el modelo de calidad FURPS.

## CAPITULO II

### 2. Marco Teórico

#### 2.1 Calidad

Se define como la propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor. (Española, 2008)

La International Standards Organization, ISO en la norma 8402:1994, define a la calidad como la totalidad de propiedades y características de un producto, proceso que le concede su aptitud para cubrir necesidades expresadas. (ISO, 2000)

##### 2.1.1 Términos relacionados con la Calidad

Según Aiteco (2016), los términos relacionados con la calidad son los siguientes:

- **Producto**

Es el resultado que se obtiene luego de haber concluido con un proceso, puede ser un servicio, una mercadería o un producto.

- **Cliente**

Persona la cual recibe, utiliza el producto o servicio sea este gratuito o pagado. En esta misma línea, se habla de cliente interno cuando el receptor es un miembro de la misma organización; y de cliente externo en el caso de que no pertenezca a la organización.

- **Características del producto**

Son las propiedades que debe cumplir el producto o servicio con las cuales se espera satisfacer las necesidades del cliente.

Pueden ser de distinto tipo, según la naturaleza del producto y de las necesidades que se quieren satisfacer.

- **Necesidades del cliente**

Requerimiento por parte del individuo que le motiva a la acción para su satisfacción. Las características del producto son las que deben satisfacer las necesidades del cliente.

## 2.2 Calidad de Software

El software en la actualidad se ha convertido en herramienta esenciales y estratégicas para cumplir con los objetivos organizacionales, por tal razón el interés por la calidad del software crece en la medida que los usuarios son más exigentes y requieren de más atributos de calidad entre ellos: rendimiento, usabilidad, seguridad, portabilidad, mantenibilidad y funcionalidad. (Iti, 2019)

La IEEE (1990), manifiesta que es el nivel con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades del cliente.

Pressman (1998), afirma que la calidad es la concordancia del software producido con los requerimientos explícitamente establecidos con los estándares de desarrollo prefijados y con los requerimientos implícitos no establecidos formalmente, que desea el usuario.

El usuario final es quien tiene la última palabra sobre la calidad del software pues su percepción estará definida por lo que tenga o no el software.

Para garantizar la calidad de software es importante implementar algún modelo o estándar de calidad que permita la gestión de atributos en el proceso de construcción de software, teniendo en cuenta que la concordancia de los requisitos y su construcción son la base de las medidas de calidad establecidas.

## 2.3 Atributos de Calidad

Son cualidades o propiedades de calidad que debe cumplir la aplicación, estas se miden en función de sus requisitos no funcionales.

Para facilitar su medición durante la verificación, debe expresarse cuantitativa o cualitativamente.

Según Brey (2008), existe una gran variedad de atributos que se pueden medir dependiendo del tipo de aplicación y su funcionalidad, los más comunes son:

- **Simplicidad:** hace referencia a la ausencia de complejidad o dificultades para realizar el manejo de la aplicación.
- **Rendimiento:** es el grado en que un sistema o componente lleva a cabo una funcionalidad específica dada una restricción de velocidad, precisión, etc. Y el uso eficiente de los recursos. (Hernandez, 2017)



- **Disponibilidad:** es el grado de operabilidad de un sistema o componente.  
Dentro de la disponibilidad, generalmente se agrupa los siguientes atributos:
  - Confiabilidad
  - Recuperación a fallos
  - Tolerancia a fallos
- **Seguridad:** es la medida sobre la habilidad del sistema para resistirse al uso por parte de usuarios no autorizados mientras sigue proveyendo servicio a sus usuarios legítimos. (Álvaro Javier, 2016)  
Otra de sus funciones es determinar las personas específicas que puedan hacer uso de la información o modificación.
- **Usabilidad:** es el grado de facilidad del sistema para ser operado por el usuario.
- **Portabilidad:** hace referencia a la destreza del software para ser transferido de un ambiente a otro y funcionar sin ningún problema.

## 2.4 Modelos de Calidad

Según Álvarez (2017), son herramientas que guían a las Organizaciones a la mejora continua y la competitividad dándoles especificaciones de qué tipo de requisitos debe implementar para poder brindar productos y servicios de alto nivel.

Los modelos de calidad permiten:

- Definición estructurada de criterios de evaluación
- Especificación de requisitos con relación a ellos
- Descripción de componentes en un marco común.
- Definición de métricas y prioridades

## 2.5 Modelos de Calidad de Software

Es un conjunto de buenas prácticas para el ciclo de vida del software, enfocado en los procesos de gestión y desarrollo de proyectos. Construir un modelo de calidad es bastante complejo y es usual que estos modelos descompongan la calidad del producto software jerárquicamente en una serie de características y sub-características que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad. (Leidy, 2012)

## 2.6 Modelo FURPS

Modelo de calidad de software desarrollado por Hewlett-Packard en el año 1987 en el que se desarrollan un conjunto de factores de calidad de software, bajo el acrónimo de FURPS. Los criterios son: funcionalidad, usabilidad, confiabilidad, desempeño y capacidad de soporte. (Constanzo, 2014)

Este modelo dispone de una serie de pruebas para las diferentes etapas del producto, los usuarios prueban el producto antes de comercializarlo y obtener un “lead-back”. Asimismo, existe un plan de soporte definido que incluye una base de datos con todos los errores registrados para poder subsanar los errores. (Reynol Solórzano Pérez, 2016)

FURPS se aplica realizando los siguientes pasos:

- Asignación de prioridades
- Definición de los atributos de calidad que pueden ser medidos.

En cuanto al atributo de rendimiento de acuerdo con el modelo FURPS menciona que se debe medir los siguientes indicadores velocidad de procesamiento, tiempo de respuesta y consumo de recursos. (Lozano, 2015)

### 2.6.1 Rendimiento Web

Es la medición del tiempo que transcurre desde el momento en que se ingresa a un sitio o a una aplicación web determinada y esta se muestra en su totalidad, en esta petición a través del Protocolo de Transferencia de Hipertexto (HTTP) pueden verse afectados los recursos que usa el servidor de aplicaciones. (Solvetic, 2015)

- **Eficacia:** capacidad del sistema para poder completar las tareas asignadas y cumplir con un objetivo propuesto.
- **Tiempo de Respuesta:** tiempo establecido para completar una tarea, incluyendo los accesos a la memoria RAM, el disco, las actividades de E/S y los consumos del S.O. Generalmente el tiempo máximo de respuesta de un sistema es de 5sg para ser eficiente.
- **Utilización de Recursos:** hace referencia a la cantidad de recursos software necesarios para el correcto funcionamiento del sistema.

**Tabla 1.** Criterios asociados a factores de calidad FURPS.

<b>Sigla</b>	<b>Tipo de Requerimiento</b>		<b>Descripción</b>
F	Funtional	Funcional	Características.
			Capacidades.
			Aspectos de seguridad.
U	Usability	Facilidad de Uso	Factores Humanos.
			Ayuda.
			Documentación.
R	Reliability	Fiabilidad	Frecuencia de fallos.
			Capacidad de recuperación de fallo.
			Grado de previsión.
P	Performance	Rendimiento	Eficacia
			Velocidad de procesamiento.
			Tiempo de respuesta.
			Utilización de recursos.
S	Supportability	Soporte	Adaptabilidad.
			Mantenimiento.
			Facilidad de configuración.
+	Plus	Implementación	Limitaciones de recursos.
			Lenguajes y herramientas.
			Hardware.
		Interfaz	Restricciones impuestas para la interacción con el sistema.
		Operaciones	Gestión del sistema.
			Pautas administrativas.
			Puesta en marcha.
		Empaquetamiento	Forma de distribución.
		Legales	Licencias.
			Derechos de autor.

**Fuente:** Constanzo (2014).

### 2.6.2 Valores de ponderación del rendimiento según el modelo FURPS

Según Constanzo (2014) y Vivanco (2017), establecen que la dimensión de ponderación establecidas por el modelo de calidad de software FURPS son aquellas que muestran en la Tabla 2.

**Tabla 2:** Valores de ponderación establecidos según el modelo FURPS.

<b>Criterios a Evaluar</b>	<b>Dimensión de ponderación según FURPS</b>
Eficacia	95%
Tiempo de Respuesta	5sg
Utilización de Recursos	25%

### 2.6.3 Ventajas y Desventajas del Modelo FURPS

Según Pressman (2002), en su libro Ingeniería de Software, un enfoque práctico señala que las ventajas y desventajas del modelo FURPS son aquellas que se describen a continuación:

#### Ventajas

- Permite encontrar, evaluar y reducir errores que se presentan en el producto.
- Se pueden reducir los riesgos de no considerar alguna de las facetas del desarrollo de un sistema.
- Es posible estandarizar algunos criterios para poder obtener los requerimientos.
- Los criterios son de fácil comprensión lo que facilita su implementación.

#### Desventajas

- El modelo tiene bastante métricas lo que puede generar mayor demanda en tiempo y costos.

## 2.7 Diseño de Software

### 2.7.1 Modelo Cascada

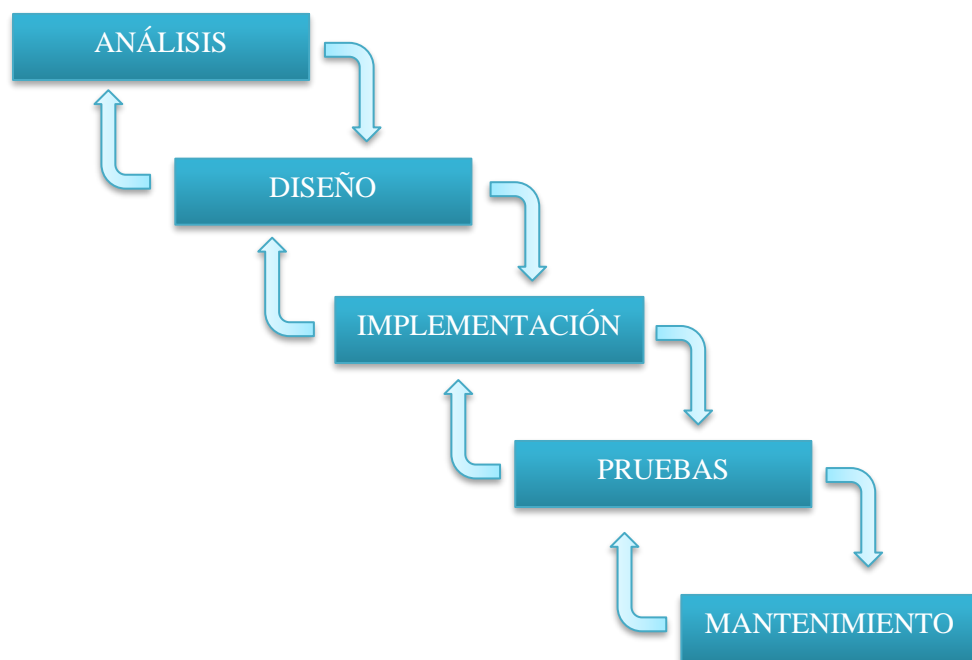
Proceso de desarrollo secuencial, se concibe como un conjunto de etapas que se ejecutan una tras otra. Se denomina así por las posiciones que ocupan las diferentes fases que componen el proyecto, colocadas una detrás de la otra, y siguiendo un flujo de construcción de arriba hacia abajo, como una cascada. (Domínguez, 2016)

#### Fases

Según Gutierrez (2011), las fases del modelo en cascada son las siguientes:

- **Análisis y definición de requerimientos:** se establecen los objetivos y restricciones establecidos por el usuario que usara el sistema.
- **Diseño del sistema:** se define la arquitectura del sistema.

- **Implementación:** se desarrolla el sistema y se realiza las pruebas unitarias para comprobar que cada parte cumple con las especificaciones.
- **Pruebas:** se comprueba que se cumpla con los requisitos de usuario establecidos previamente.
- **Mantenimiento:** es la instalación del sistema, operación y capacitación a los usuarios del sistema, en esta fase se realiza la corrección de errores y la implementación de mejoras.



**Figura 1:** Ciclo de vida metodología cascada (Sari, 2017).

### **Ventajas y Desventajas del modelo Cascada**

Según Pons (2010), las ventajas y desventajas son las siguientes:

#### **Ventajas**

- Admite interacciones, permite volver a una etapa anterior del proyecto.
- Planificación sencilla.
- La actividad puede descomponerse de tal forma que una fase no necesite de otra fase.

## Desventajas

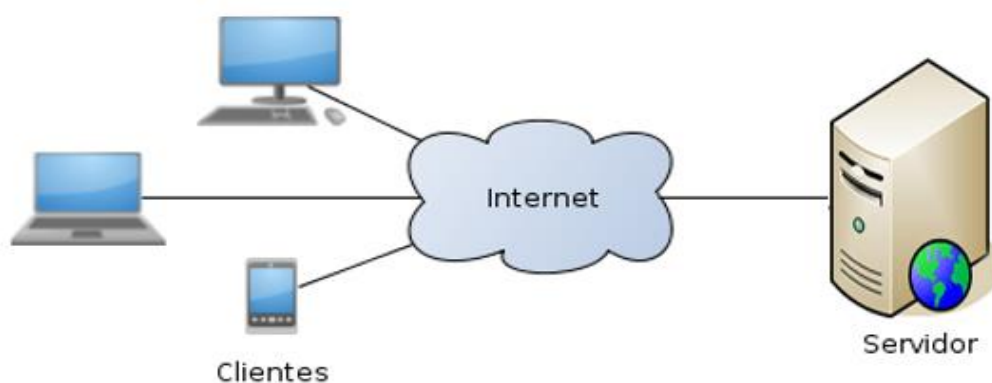
- Si se ha cometido errores y no se detectan en la etapa inmediatamente siguiente, es costoso y difícil volver atrás para realizar la corrección.
- Los resultados no se ven hasta las etapas finales del ciclo.
- Retardo en entregar partes del producto.

### 2.7.2 Arquitectura cliente-servidor

Según Marini (2012), es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores o servidores, y los solicitantes llamados clientes. Los clientes ejecutan peticiones y el servidor proporciona la respuesta.

La arquitectura cliente- servidor está compuesta de la siguiente manera:

- **Red:** Es un conjunto de clientes, servidores y base de datos unidos de una manera física o no física en el que existen protocolos de transmisión de información establecidos.
- **Cliente:** Es el encargado de realizar las peticiones al servidor mediante una interfaz gráfica.
- **Servidor:** Esperan las peticiones realizadas por los clientes para procesarla y enviarle al cliente la respuesta.
- **Base de datos:** Colección de información organizada de tal manera que sea fácilmente accesible, gestionada y actualizada tanto para los servidores como para los usuarios.



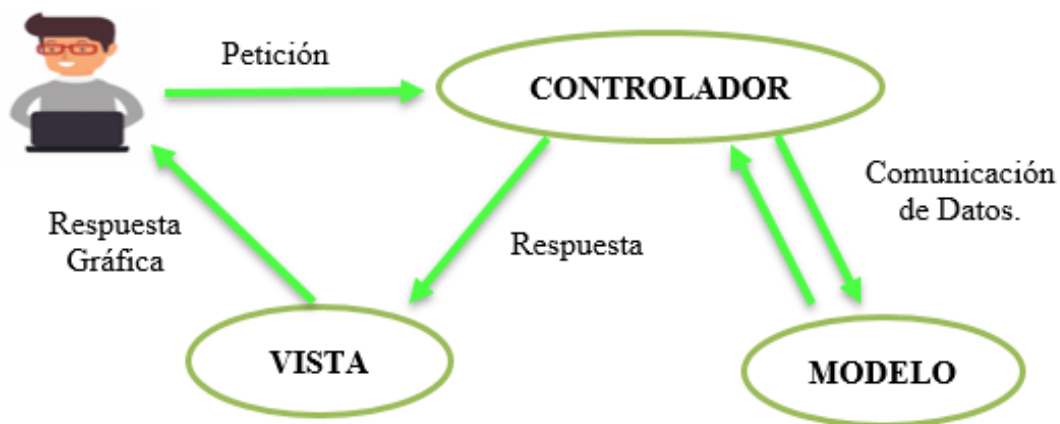
**Figura 2:**Arquitectura cliente- servidor.

### 2.7.3 MVC

Según Fernández (2012), 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 esta arquitectura existe un sistema central que gestiona las entradas y la salida del sistema, uno o varios modelos que se encargan de averiguar los datos e información necesaria y la interfaz que muestra los resultados al usuario final.

Este modelo según Hernández (2015), está dividido en las siguientes capas:

- **Modelo:** Se encarga de manipular, gestionar y actualizar los datos.
- **Vista:** Se encarga de recibir y mostrar los datos al usuario.
- **Controlador:** se encarga de gestionar las instrucciones que recibe, atiende y las procesa.



**Figura 3:** Modelo vista controlador.

## 2.8 Desarrollo de Software

### 2.8.1 MySQL

Es el sistema gestor de bases de datos relacional por excelencia.

Es un SGBD multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Además, es el más usado en aplicaciones creadas como software libre.

Se ofrece bajo la GNU GPL, aunque también es posible adquirir una licencia para empresas que quieran incorporarlo en productos privativos.

Las principales ventajas de este Sistema Gestor de Bases de datos son:

- Facilidad de uso y gran rendimiento
- Facilidad para instalar y configurar
- Soporte multiplataforma
- Soporte SSL

La principal desventaja es la escalabilidad, es decir, no trabaja de manera eficiente con bases de datos muy grandes que superan un determinado tamaño. (Rivera, 2008)



**Figura 4:** MySQL.

### **2.8.2 NetBeans**

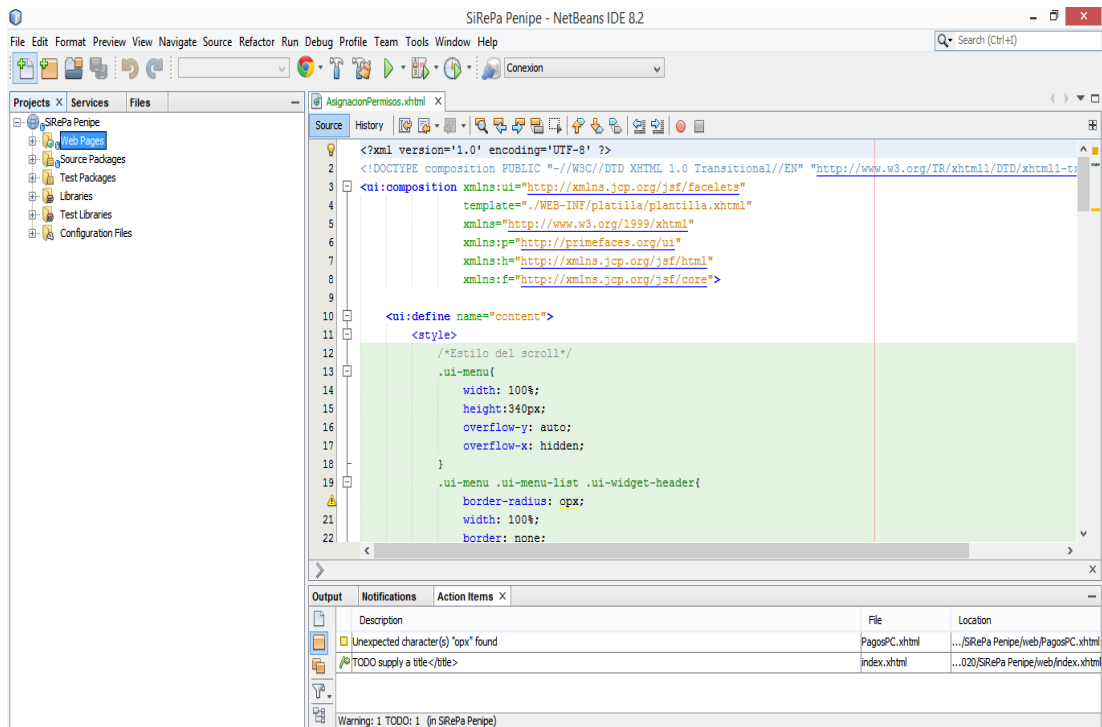
Es un entorno integrado de desarrollo que permite realizar todas las tareas asociadas a la programación como editar código, compilarlo, ejecutarlo y depurarlo. Es utilizado por diferentes programadores para crear aplicaciones de escritorio y web, es multilinguaje, de código abierto sin restricciones de uso. (Gimeno, 2011)

#### **Características**

Según Geovanny (2015), las características son las siguientes:

- Multilinguaje
- De código abierto
- Gratuito
- Control de versiones.
- Soporte para JavaScript
- Herramienta modular de desarrollo.
- Incluye servicios para el control del interfaz de usuario.





**Figura 5:** Escenario de trabajo en NetBeans.

### 2.8.3 JSF

Según Guzmán (2016), Java Server Faces es una tecnología y framework del lado del servidor para crear interfaces de usuario para aplicaciones web usando el lenguaje de programación Java. Está basada en el patrón Modelo Vista Controlador.

#### Ventajas

Según Gonzales (2019), las principales ventajas son las siguientes:

- El código JSF crea las vistas (etiquetas JSP) es muy parecido al HTML estándar. Lo pueden utilizar fácilmente desarrolladores y diseñadores web.
- JSF se integra dentro de la página JSP y se encarga de la recogida y generación de los valores de los elementos de la página.
- JSF permite introducir JavaScript en la página, para acelerar la respuesta de la interfaz en el cliente.
- JSF es extensible, puede desarrollar nuevos componentes a la medida. También permite modificar el comportamiento del framework mediante APIs que controlan su correcto funcionamiento.

## 2.9 Aplicaciones para medición del rendimiento

### 2.9.1 JMeter

Según Apache (2019), es una aplicación Java 100% pura diseñada para cargar el comportamiento funcional de las pruebas y medir el rendimiento a las aplicaciones web.

### 2.9.2 LoadRunner

Es una aplicación de prueba de software de Micro Focus. Utilizada con el fin de probar aplicaciones, medir el comportamiento del sistema y el rendimiento bajo carga. Puede simular miles de usuarios utilizando el software simultáneamente. (MicroFocus, 2017)

### 2.9.3 Programas para medición del rendimiento

**Tabla 3:** Comparación de JMeter y LoadRunner.

Características	Programas	
	JMeter	LoadRunner
Servicios REST	Si	Si
Servicios SOAP	Si	Si
Métricas de Rendimiento	Si	Si
Precio	Gratis	Hasta 50 users
Generación de Carga	Ilimitada	Limitada
Compatible con base de datos JDBC	Si	Si
Integración con Plugins	Si	No
Núcleo altamente extensible	Si	No
Visualización de Datos	Fácil de entender	Formato sin procesar
Análisis de datos y reportes	Si	No
Sistemas operativos compatibles	Windows, Mac, Linux	Windows, Linux
Open Source	Si	No

**Fuente:** Pérez (2015).

En la Tabla 3 se puede apreciar la comparación de las herramientas de rendimiento en la cual se puede observar que JMeter es el más completo, puesto que permite la integración con plugins, compatibilidad con JDBC, los datos analizados son más fáciles de entender, la generación de la carga es ilimitada, licencia gratuita y sobre todo permite medir el rendimiento por lo cual en esta investigación se utilizara JMeter.

## 2.10 Metodología para la evaluación del rendimiento con JMeter

Según Apache (2019), la metodología de las pruebas de rendimiento consiste en realizar las siguientes actividades:

- **Definir los escenarios que se va a evaluar**

Se define todas las acciones que van a realizar los usuarios en el sitio web, como visitar alguna página principal y pulsar en algún enlace que conduzca a otra página de esta forma se obtendrá una medida clara de los límites que podrá soportar nuestro sitio web.

- **Generar un script de navegación**

Mediante JMeter se puede configurar un proxy para generar este script simplemente visitando las páginas con el navegador.

- **Abrir script con JMeter**

Una vez generado el script se procede a abrir el fichero generado anteriormente.

- **Añadir un receptor**

Se debe añadir algún mecanismo para obtener los datos mediante JMeter. Lo recomendable es añadir un Informe Agregado.

- **Definir el número de iteraciones**

A medida que se realice más interacciones los datos se aproximarán más a la realidad, en este apartado se obtendrán tiempos medios de respuesta.

- **Definir el número de usuarios concurrentes**

El número de hilos representa el número de procesos que “visualizarán” páginas web. El número de muestras será igual a este valor multiplicado por el número de iteraciones.

- **Obtener el informe de rendimiento**

Luego de haber concluido con las indicaciones anteriores se procede a ejecutar el programa a través del cual se observarán los valores recogidos en el informe agregado.

## CAPITULO III

### 3 Metodología

#### 3.1 Tipo y diseño de la investigación

La presente investigación se basó en el método cuantitativo, se midió el rendimiento según los criterios establecidos en el modelo de calidad de software FURPS.

Basado en un estudio longitudinal, se recopilan datos después de varias pruebas durante un periodo de tiempo, permitiendo examinar las variaciones que presentan las variables de rendimiento.

Para analizar y explicar los resultados de la investigación se hizo uso del método analítico, permitió desglosar cada uno de los criterios de rendimiento que se midieron para posteriormente emitir una conclusión de acuerdo con los datos obtenidos.

#### 3.2 Según el objeto de estudio

- **Investigación de Campo:** A través de esta investigación se pudo extraer todos los requerimientos del sistema de recaudación de patentes para posteriormente evaluar su rendimiento.
- **Investigación Bibliográfica:** Se realizó la recolección de información relacionada al método de calidad de software FURPS utilizando técnicas y estrategias para acceder a documentos como: tesis, libros y artículos que ayudaran a guiar la investigación.

#### 3.3 Población de estudio y tamaño de la muestra

En la investigación se establece como población a los funcionarios del GADM Penipe con un total de 75 funcionarios y como muestra el personal que labora en el área de tecnologías de la información, avalúos y catastros, recaudación y contabilidad, con un total de 10 funcionarios.

#### 3.4 Técnicas de recolección de Datos

Para la recopilación de información se aplicó la técnica de la observación y la entrevista, posteriormente con la ayuda de la herramienta JMeter se procedió a recolectar los datos referentes a la medición de los criterios de rendimiento establecidos por el modelo de calidad de software FURPS.

### 3.4.1 Observación

Esta técnica ayudó de manera inicial para identificar el problema que presenta la institución, se identificó que el GAD Penipe no cuenta con un sistema de recaudación de patentes que garantice el salvaguardo de información, un correcto procesamiento de datos y sobre todo agilidad en los procesos de cobranza de patentes, ahorrando tiempo a la ciudadanía.

### 3.4.2 Entrevista

Esta técnica fue utilizada para establecer los requisitos tanto de diseño como de funcionalidad del sistema de recaudación de patentes, tras obtener toda la información necesaria se realizó la planificación para el desarrollo del presente proyecto de investigación.

## 3.5 Técnica de Análisis e interpretación de la información

### 3.5.1 Herramientas Utilizadas

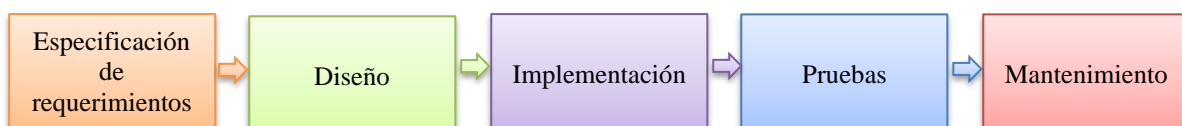
Las mediciones de rendimiento se efectuaron con la ayuda de la aplicación JMeter, esta aplicación proporciona valores numéricos relacionados a la eficacia, tiempo de respuesta y a la vez proporciona gráficas de los valores obtenidos.

## 3.6 Desarrollo de Software

Para el desarrollo del sistema web de recaudación de patentes se utilizó la metodología en cascada. Esta metodología es un proceso secuencial, cada etapa se ejecuta una a continuación de otra. Cada fase tiene objetivos y tareas bien definidos que satisfacen las necesidades del software. (Domínguez, 2016)

### Fases:

El sistema de recaudación de patentes se desarrolló cumpliendo las fases que se muestran en la figura 6.



**Figura 6:** Fases modelo cascada.

### 3.6.1 Análisis de requerimientos.

La figura 7 muestra el proceso de generación, cobro de patentes y las actividades que realiza cada uno de los usuarios del sistema.

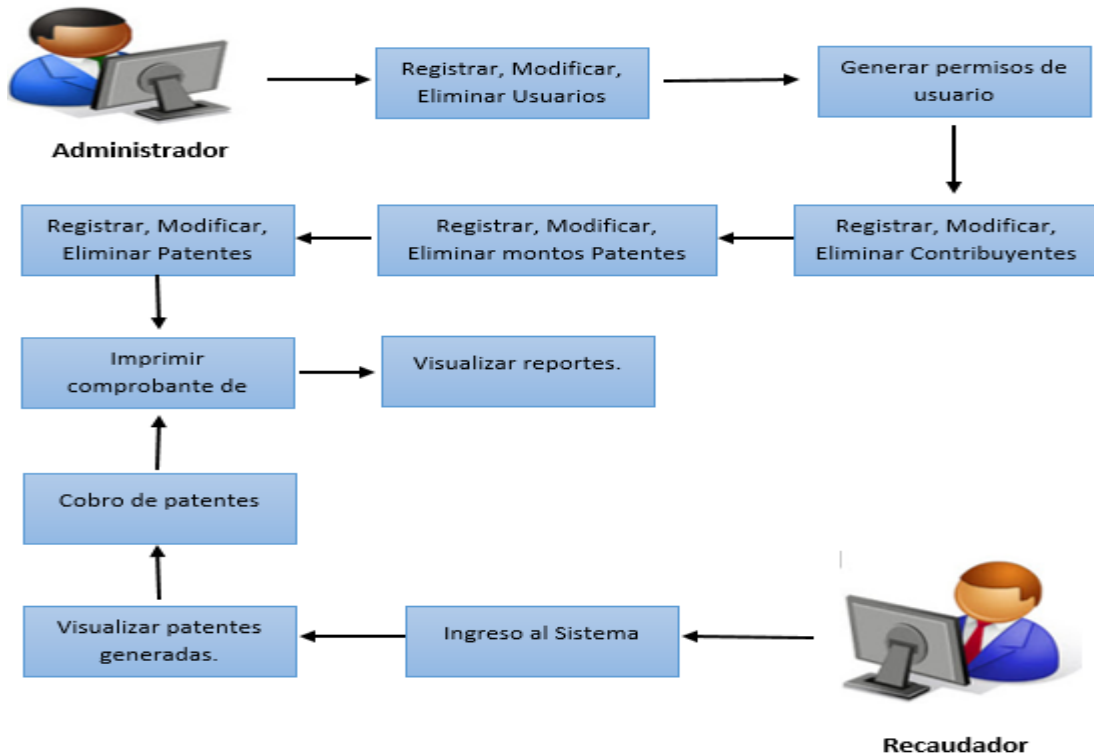


Figura 7: Flujo de Procedimiento de Patentes.

- **Requisitos Funcionales**

Tabla 4: Requisitos Funcionales del sistema.

Requisitos	Descripción
<b>Crear usuarios.</b>	El sistema deberá permitir crear diferentes cuentas de usuario con sus respectivos datos como el nombre, apellido, teléfono, cédula, usuario y contraseña. Además de guardar, actualizar y eliminar datos que se crea necesarios.
<b>Asignar permisos de usuario.</b>	El sistema permitirá habilitar ciertos módulos del sistema para cada usuario registrado en el mismo.
<b>Crear tipo de contribuyentes.</b>	El sistema permitirá crear diferentes tipos de contribuyentes tales como personas naturales, civiles, asociaciones, etc.
<b>Crear contribuyentes.</b>	El sistema deberá permitir registrar a cada contribuyente que posee algún tipo de

	negocio dentro del cantón sujeto a cancelar el pago de patente anual. Además, se podrá guardar, actualizar y eliminar datos necesarios.
<b>Crear patentes.</b>	El sistema permitirá crear la patente respectiva por cada contribuyente registrado en el sistema. Además, se podrá guardar, actualizar y eliminar datos necesarios.
<b>Generar pagos por patentes.</b>	El sistema deberá ser capaz de generar automáticamente la carta de pago anual de cada patente registrada.
<b>Visualizar deuda de patente por cada contribuyente.</b>	El sistema deberá ser capaz de mostrar la cantidad de deuda con la que cuenta cada contribuyente a la fecha actual de la consulta.
<b>Imprimir Comprobantes.</b>	El sistema permitirá generar el comprobante de pago una vez cancelado el valor de la patente y posteriormente imprimirlo.
<b>Visualizar Reportes.</b>	El sistema permitirá visualizar reportes tanto de valores recaudados como de los valores por acreditar durante el año en curso.

- **Requisitos No Funcionales**

**Tabla 5:** Requisitos No Funcionales del sistema.

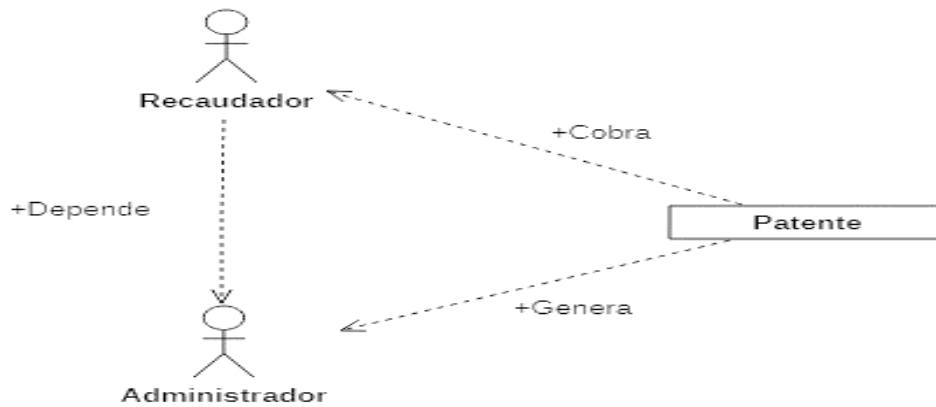
<b>Requisitos</b>	<b>Descripción</b>
<b>Usabilidad</b>	El sistema debe ser fácil de utilizar, aprender, entender y ser atractivo para el usuario.
<b>Funcionalidad</b>	El sistema deberá responder a las peticiones solicitadas según sea el caso.
<b>Rendimiento</b>	El sistema deberá ser capaz de responder a las peticiones realizadas en el menor tiempo posible.
<b>Seguridad</b>	El sistema deberá salvaguardar su información por tal motivo cuenta con un sistema de logueo que garantiza el acceso solamente a usuarios autorizados.
<b>Disponibilidad</b>	El sistema debe permitir acceder a sus usuarios las 24 horas del día y todos los días del año.

### 3.6.2 Diagramación

#### Diagrama de casos de uso

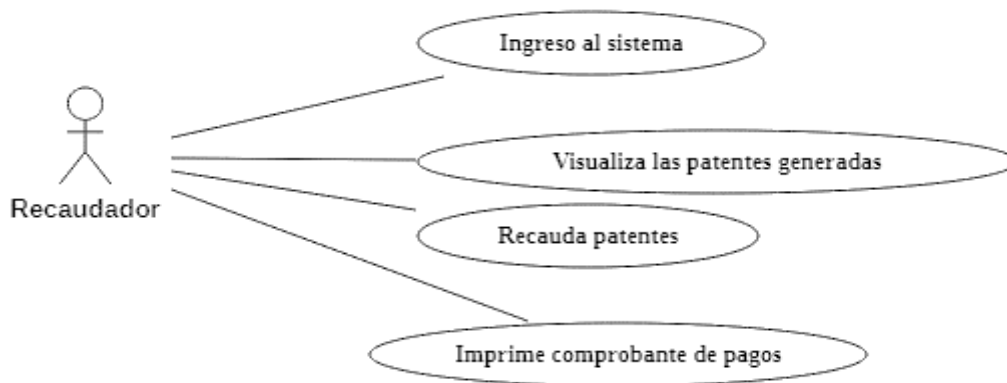
Muestra las funcionalidades y usos que cada usuario tendrá sobre el sistema de recaudación de patentes.

#### Actores



**Figura 8:** Caso de uso Sistema de recaudación de patentes.

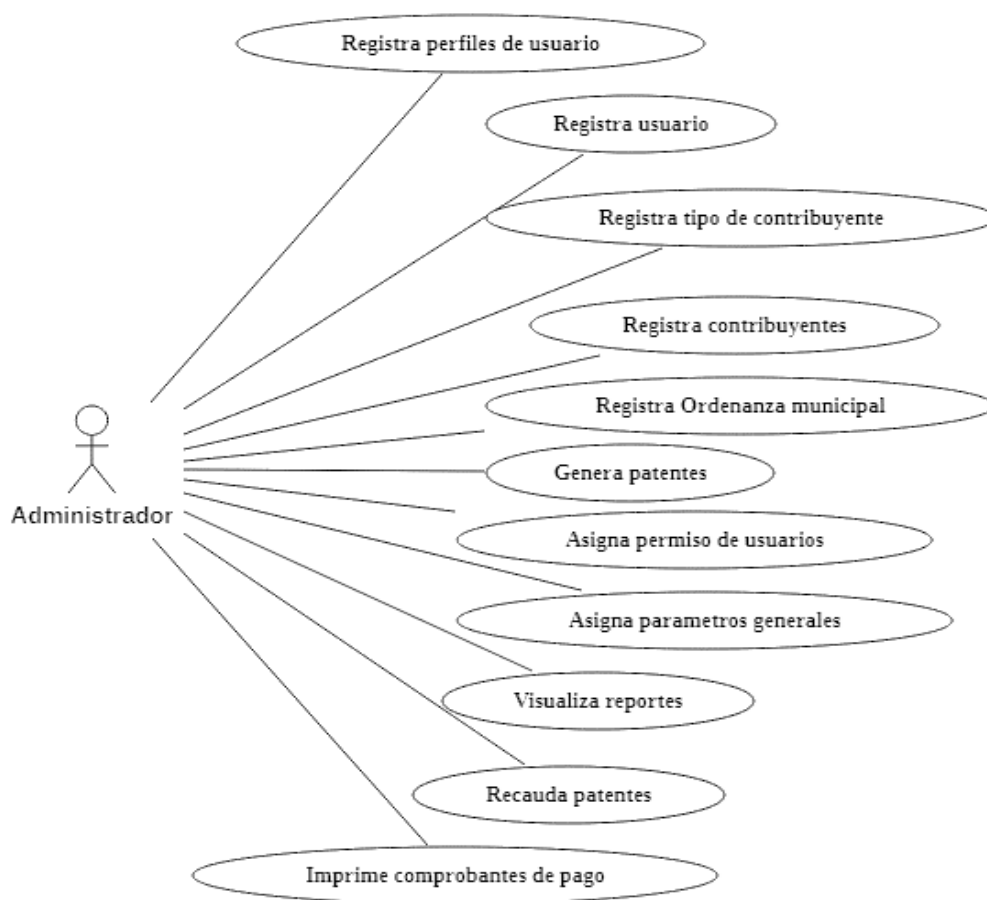
#### Recaudador



**Figura 9:** Caso de uso Recaudador.



## Administrador



**Figura 10:** Caso de uso Administrador.

### 3.6.3 Diseño

- **Vista Conceptual**

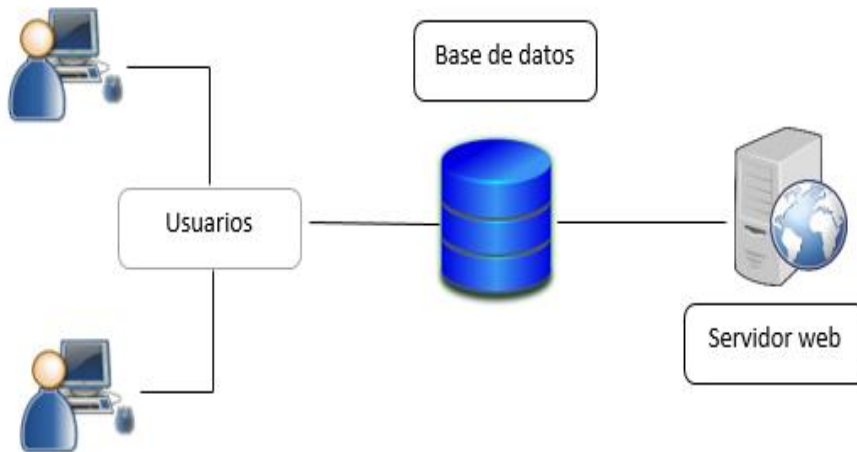
El sistema web de recaudación de patentes esta administrado por dos usuarios: Administrador y Recaudador.

- **Administrador:** Posee el control total del sistema de recaudación de patentes.
- **Recaudador:** Realiza las funciones de visualización de patentes generadas de acuerdo con cada contribuyente, el cobro de patentes y la impresión del comprobante de pago relacionada a cada patente.



**Figura 11:** Vista Conceptual.

- **Vista Física**



**Figura 12:** Vista Física.

### 3.7 Implementación

#### Diagrama de clases

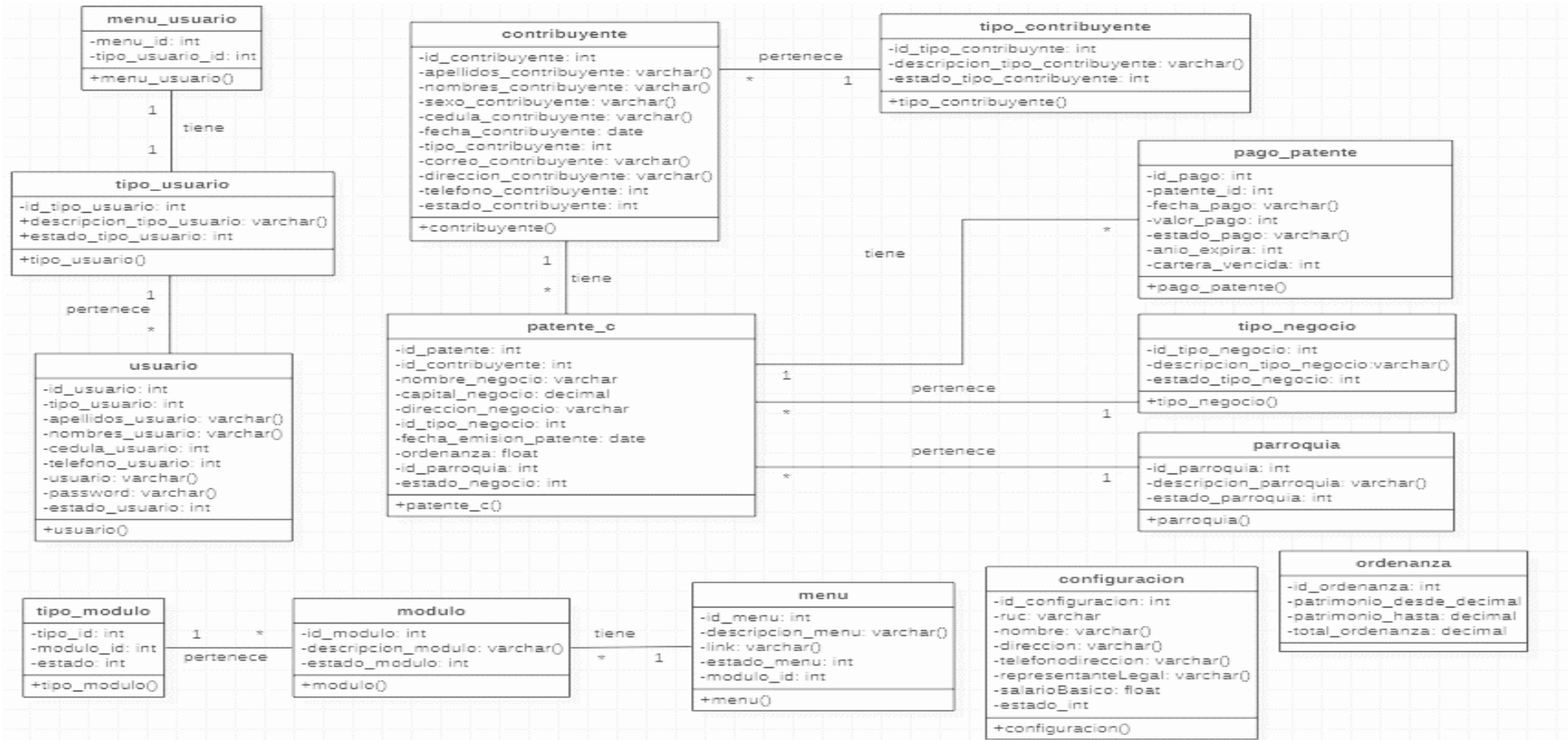


Figura 13: Diagrama de clases.

## Diagrama de bases de datos

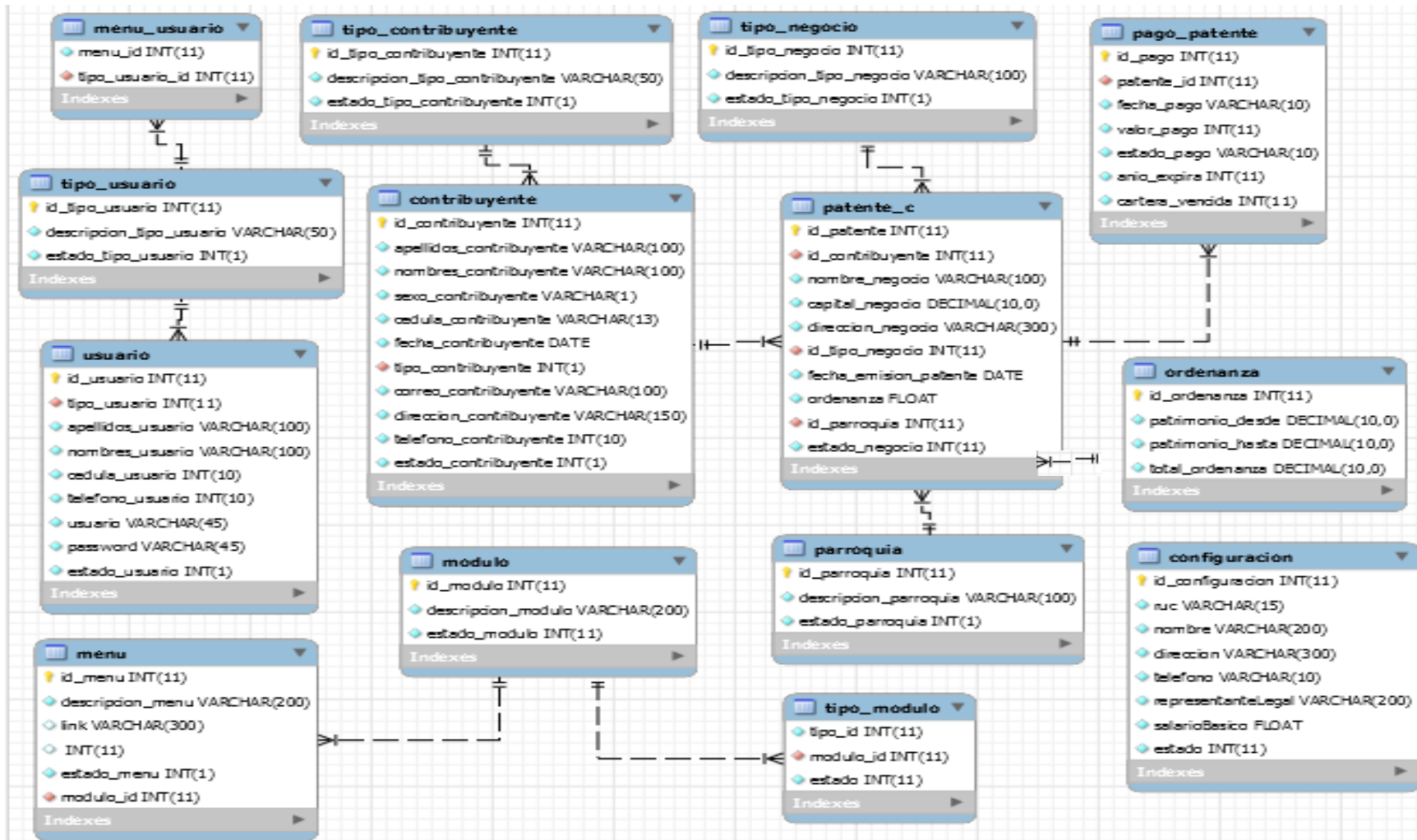


Figura 14: Diagrama de base de datos.

### **3.8 Pruebas**

Concluido con el desarrollo de la aplicación se realizó las pruebas de todo los módulos y componentes del sistema web de recaudación de patentes.

### **3.9 Mantenimiento**

Al momento de probar el sistema se realizaron cambios por los errores existentes. Los cambios realizados fueron funcionales y de diseño, estos se aplicaron cuidando que no afecte a la integridad del sistema de recaudación de patentes.

## CAPITULO IV

### 4 Resultados y Discusión

Como parte de la investigación, se realizó las pruebas de rendimiento del sistema de recaudación de patente, mediante la ayuda de la aplicación JMeter y aplicando los criterios de rendimiento establecidos por el modelo de calidad de software FURPS.

El número identificado de casos de prueba para el sistema de recaudación de patentes fue 10 funcionarios del GAD Penipe, estos usuarios son las personas encargadas de hacer uso del sistema dentro del GAD Municipal.

**Tabla 6:** Casos de Prueba.

<b>Resultados Casos de Prueba Estado Inicial</b>		
<b>Ejecutados</b>	<b>Exitosos</b>	<b>Fallidos</b>
10	10	0

En primera instancia como se observa en la tabla 6, los datos recopilados de las pruebas de rendimiento a 10 usuarios, se demuestra que el 100% de casos fue exitoso



**Figura 15:** Resultado de pruebas Iniciales.

Las pruebas en un estado inicial fueron ejecutadas de manera exitosa, cada una de estas esta detallado en el ANEXO 1.

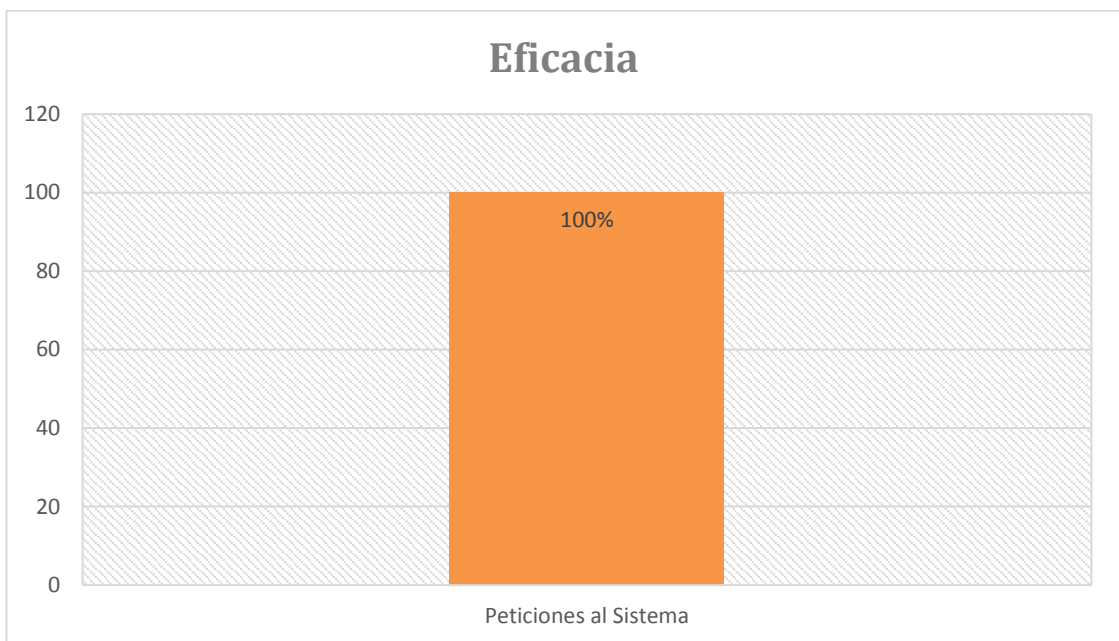
#### 4.1 Análisis de los indicadores

La tabla 7 y figura 16 muestran la medición de la eficacia realizada al sistema de recaudación de patentes, obteniendo un porcentaje del 100% de peticiones efectuadas sin margen de error. Las peticiones realizadas se pueden visualizar de manera detallada en el ANEXO II.

##### 4.1.1 Dimensión eficacia

**Tabla 7:** Dimensión Eficacia.

Dimensión	Indicador	Petición
Eficacia	Número de peticiones realizadas correctamente	140
	Promedio (%)	100%



**Figura 16:** Dimensión Eficacia.

##### 4.1.2 Tiempo de Respuesta

Luego de realizar las respectivas pruebas que se detallan en el ANEXO III, la figura 11 muestra el promedio en milisegundos del tiempo de respuesta del total de interacciones realizadas al sistema, con un número de 10 usuarios conectados simultáneamente. Estos datos se obtuvieron mediante la fórmula para calcular el promedio del tiempo de respuesta que se muestra a continuación.

### Fórmula para calcular el Promedio

$$\bar{n} = \frac{1}{2} \sum_{i=1}^n a = \frac{a1 + a2 + a3 + a4 + a5 + \dots + an}{n}$$



**Figura 17:** Tiempo de Respuesta.

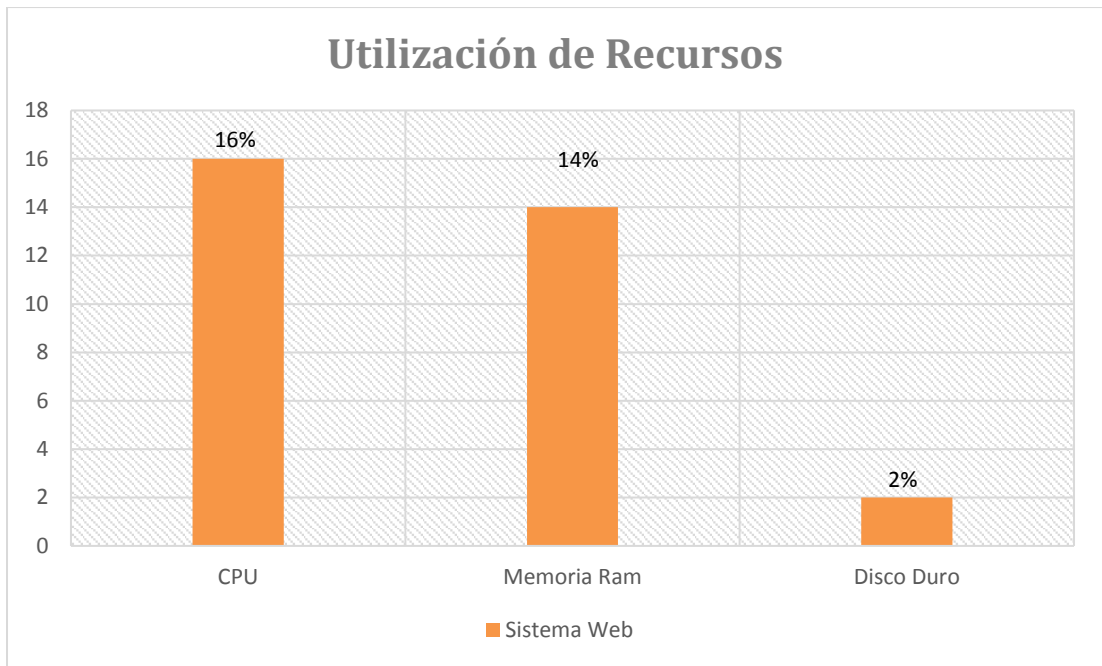
### 4.1.3 Utilización de Recursos

Una vez realizadas las respectivas pruebas al sistema de recaudación de patentes, en la tabla 8 se observa el porcentaje de los indicadores (consumo de recursos).

**Tabla 8:** Utilización de recursos.

<b>Dimensión</b>	<b>Indicador</b>	<b>Sistema Web GAD Penipe</b>
Consumo de Recursos	Promedio de uso de CPU	16%
	Promedio de uso de Memoria RAM %	14%
	Promedio de uso de Disco Duro %	2%





**Tabla 9:** Comparación de valores establecidos por FURPS y valores obtenidos en las pruebas

<b>Criterios a Evaluar</b>	<b>Dimensión de ponderación según FURPS</b>	<b>Dimensiones obtenidas</b>
Eficacia	95%	100%
Tiempo de Respuesta	5sg	993,5 ms
Utilización de Recursos	25%	10.66%

## CONCLUSIONES

- Los factores de calidad de software establecidos en el modelo FURPS para analizar la calidad del software son: funcionalidad, usabilidad, confiabilidad, rendimiento y soporte.
- Los indicadores del factor de rendimiento basados en el modelo FURPS son: peticiones HTTP realizadas correctamente, promedio de tiempo de respuesta y uso de recursos como el disco duro, la memoria y el CPU. Para obtener estos valores se utilizó la aplicación JMeter.
- El desarrollo del sistema para la recaudación de patentes se realizó bajo el modelo de desarrollo de software cascada, con una arquitectura cliente-servidor y las herramientas IDE java-NetBeans, lenguaje de programación Java y como gestor de base de datos MySQL.
- Para comprobar el rendimiento según el modelo FURPS, se realizó 10 peticiones HTTP realizadas al sistema. Todas las peticiones se realizaron de manera adecuada cumpliendo al 100%, el tiempo de respuesta en cada una de las interacciones no sobrepasa de los 5 segundos establecidos en el modelo FURPS. El promedio total del tiempo de respuesta del sistema es de 993.5 ms. El uso de recursos hace referencia a la cantidad de espacio utilizado en el CPU con un 16% de su capacidad total, memoria RAM 14% de uso y el disco duro 2%. Se concluye que el sistema cumple con el factor de calidad de rendimiento establecido por el modelo FURPS.

## **RECOMENDACIONES**

- FURPS es un modelo de calidad de software que establece varios factores con sus respectivas características para garantizar la calidad del producto, por tal motivo se recomienda estudiar y analizar profundamente cada uno de sus factores para poder establecer qué criterios se va a analizar y medir.
- Se recomienda antes de iniciar con el desarrollo del sistema web establecer cuáles son los requisitos funcionales y escoger las herramientas adecuadas para el desarrollo.
- Para la medición de pruebas de rendimiento en sistemas web se recomienda utilizar JMeter y el modelo de calidad de software FURPS. Este modelo establece criterios que garantiza un software de calidad.

## 5 BIBLIOGRAFIA

- Aiteco. (2016). *Concepto de calidad*. Obtenido de <https://www.aiteco.com/concepto-de-calidad/>
- Alicia, R. (2011). *Aplicaciones Web*. Paraninfo. Obtenido de <https://www.neosoft.es/blog/que-es-una-aplicacion-web/>
- Álvarez, A. (2017). *Calidad Software Modelos*. Obtenido de <http://www.scielo.org.co/pdf/entra/v13n1/1900-3803-entra-13-01-00236.pdf>
- Álvaro Javier, L. P. (2016). *Comparación de dos tecnologías de desarrollo de aplicaciones móviles desde la perspectiva del rendimiento como atributo de calidad*. Obtenido de <http://revistas.utp.edu.co/index.php/revistaciencia/article/view/9278/6301>
- Apache. (22 de 10 de 2019). *Apache JMeter*. Obtenido de <https://jmeter.apache.org/>
- Batallas, H. (2013). *Desafío para los Gobiernos Autónomos*. Obtenido de <http://repositorio.uasb.edu.ec/bitstream/10644/4225/1/03-TC-Batallas.pdf>
- Brey, G. (2008). *Atributos de Calidad*. Obtenido de [http://apit.wdfiles.com/local--files/start/02\\_apit\\_atributos\\_de\\_calidad.pdf](http://apit.wdfiles.com/local--files/start/02_apit_atributos_de_calidad.pdf)
- Claudia Pons, R. G. (2010). *Desarrollo de software dirigido por modelos*. Obtenido de [http://sedici.unlp.edu.ar/bitstream/handle/10915/26667/Documento\\_completo\\_\\_\\_.pdf?squence=1&isAllowed=y](http://sedici.unlp.edu.ar/bitstream/handle/10915/26667/Documento_completo___.pdf?squence=1&isAllowed=y)
- Constanzo, M. (2014). *Modelos de Calidad de Software*. Obtenido de <http://clases3ggingsof.wikifoundry.com/page/FURPS>
- Domínguez, P. (2 de junio de 2016). *Gestión de proyectos de Desarrollo*. Obtenido de <https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>
- Española, R. A. (Mayo de 2008). *Calidad*. Obtenido de <http://buscon.rae.es/draeI/html/cabecera.htm>
- Garita-Araya, R. A. (Julio de 2013). *El desarrollo de aplicaciones y la Implementación de sistemas web en la actualidad se han convertido en la base tecnológica*. Obtenido de <http://revistaebci.ucr.ac.cr/>
- Geovanny, M. G. (2015). *Herramienta de Desarrollo Netbeans*. Obtenido de [https://www.consultorjava.com/wp/wp-content/uploads/2015/09/herramienta\\_desarrollo\\_netbeans.pdf](https://www.consultorjava.com/wp/wp-content/uploads/2015/09/herramienta_desarrollo_netbeans.pdf)
- Gimeno, J. M. (2011). *Introducción a Netbeans*. Obtenido de [https://d1wqtxts1xzle7.cloudfront.net/33971018/1-introduccioi81n-a-netbeans.pdf?1403006101=&response-content-disposition=inline%3B+filename%3DIntroduccion\\_a\\_Netbeans.pdf&Expires=1591919833&Signature=M9EF1FR~U6H~R07AGdvlo28DaQjXbbAvzgIc7LySkOIBINIs3IjhECFQ](https://d1wqtxts1xzle7.cloudfront.net/33971018/1-introduccioi81n-a-netbeans.pdf?1403006101=&response-content-disposition=inline%3B+filename%3DIntroduccion_a_Netbeans.pdf&Expires=1591919833&Signature=M9EF1FR~U6H~R07AGdvlo28DaQjXbbAvzgIc7LySkOIBINIs3IjhECFQ)
- Gonzales, C. (26 de 03 de 2009). *Introducción a JSF Java*. Obtenido de <https://www.adictosaltrabajo.com/2009/03/26/introduccion-jsf-java/>
- Gusman, P. (2016). *JavaServer Faces.org*. Obtenido de <http://www.javaserverfaces.org/>
- Gutierrez, D. (Julio de 2011). Obtenido de <https://sites.google.com/site/proyectoadpmodelosdedesarrollo/home/modelo-en-cascada>
- Hernandez, J. (2017). *Ingeniería del Software II*. Obtenido de <https://ocw.unican.es/pluginfile.php/1408/course/section/1803/tema2-calidadSistemasSoftware.pdf>

- Hernandez, U. (2015). *MVC* . Obtenido de <https://codigofacilito.com/articulos/mvc-model-view-controller-explicado>
- ISO. (2000). *Sistemas de gestión de calidad*.
- Iti. (2019). *Investigate to Innovate*. Obtenido de <https://www.iti.es/servicios/calidad-de-software/>
- Leidy, B. (2012). *Modelo de Calidad Software*. Obtenido de <https://es.scribd.com/doc/138528080/Modelos-de-Calidad-de-Software>
- Lisandro, G. N. (2015). *Un análisis comparativo de rendimiento en aplicaciones móviles* . Junin.
- Lozano, E. A. (2015). *CUADRO COMPARATIVO DE LOS MODELOS DE CALIDAD*. Obtenido de [https://profelozano.files.wordpress.com/2015/10/eduard\\_lozano\\_cuadrocomparativo\\_actividad2\\_2.pdf](https://profelozano.files.wordpress.com/2015/10/eduard_lozano_cuadrocomparativo_actividad2_2.pdf)
- Marini, E. (2012). *El Modelo Cliente/Servidor*. Obtenido de <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>
- Martínez, J. F. (2014). *Implantación de Aplicaciones Web*. España: Ra-Ma.
- Martínez, J. F. (2015). *Implantación de aplicaciones web en entornos internet, intranet y extranet*. España: Ra-Ma.
- MicroFocus. (2017). *LoadRunner*. Obtenido de <https://www.microfocus.com/es-es/products/loadrunner-load-testing/overview>
- Reynol Solórzano Pérez, M. M. (2016). *Comparación entre modelos de calidad de software para ser utilizados en el*. Obtenido de <https://eventos.uho.edu.cu/index.php/ccm/cci2017/paper/view/882/366>
- Rivera, F. L. (2008). *Bases de datos relacionales* . Medelli.
- Riyas, I. (2015). *Características de una aplicación web*. Obtenido de <http://estudiantealdeunare3irmadj.blogspot.com/p/caracteristicas-de-aplicaciones-web.html>
- Roger S. Pressman, P. (2010). *Ingeniería del software Un enfoque práctico*. México: McGraw-Hill.
- Sari, M. (20 de 02 de 2017). *Ciclo de vida en cascada*. Obtenido de [http://agrega.juntadeandalucia.es/repositorio/20022017/6b/es-an\\_2017022012\\_9122843/51\\_ciclo\\_de\\_vida\\_clsico\\_o\\_en\\_cascada.html](http://agrega.juntadeandalucia.es/repositorio/20022017/6b/es-an_2017022012_9122843/51_ciclo_de_vida_clsico_o_en_cascada.html)
- Solvetic. (12 de Marzo de 2015). *Rendimeinto Web*. Obtenido de <https://www.solvetic.com/tutoriales/article/1502-c%C3%B3mo-medir-el-rendimiento-de-una-aplicaci%C3%B3n-web/>
- Valdés, D. P. (26 de Octubre de 2007). *Bases de datos*. Obtenido de <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Vivanco, M. (2017). *Análisis de Rendimeinto de un sistema de comunicación que utiliza un sistema de cifrado*. Obtenido de <http://dspace.esoch.edu.ec/bitstream/123456789/8088/1/20T00978.PDF>
- Yenisleidy Fernandez, Y. D. (2012). Patrón Modelo-Vista-Controlador. *Telemática*, 11.

## ANEXOS

### ANEXO I: Resultado de pruebas iniciales sistema de recaudación de patentes.

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error
Loguin	10	32	14	46	8,64	0,00%
Pagina Principal	10	38	18	48	9,57	0,00%
Registro Usuario	10	64	26	79	17,16	0,00%
Perfiles de Usuario	10	52	21	72	14,59	0,00%
Permisos Usuario	10	43	19	60	14,49	0,00%
Tipo Contribuyente	10	39	21	57	12,61	0,00%
Tipo de Negocio	10	40	18	65	17,78	0,00%
Registro Contribuyente	10	51	26	79	21,89	0,00%
Registro Ordenanza	10	42	21	75	17,41	0,00%
Registro Pago Patente	10	21	16	29	4,50	0,00%
Impresion Comprobante Pago	10	18	15	23	2,49	0,00%
Parametros Generales	10	20	16	26	2,84	0,00%
Reportes	10	15	12	19	2,27	0,00%
Registro Patente	10	52	36	71	13,58	0,00%
Total	140	38	12	79	19,40	0,00%

**Figura 18:** Reporte Resumen

### ANEXO II: Resultado medición de eficacia del sistema de recaudación de patentes.

Etiqueta	# Muestras	% Error
Reportes	10	0,00%
Registro Pago Patente	10	0,00%
Impresion Comprobante Pago	10	0,00%
Parametros Generales	10	0,00%
Registro Patente	10	0,00%
Registro Ordenanza	10	0,00%
Tipo de Negocio	10	0,00%
Registro Contribuyente	10	0,00%
Loguin	10	0,00%
Perfiles de Usuario	10	0,00%
Pagina Principal	10	0,00%
Tipo Contribuyente	10	0,00%
Registro Usuario	10	0,00%
Permisos Usuario	10	0,00%
Total	140	0,00%

**Figura 19:** Medición Eficacia

**ANEXO III: Resultado medición del tiempo de respuesta del sistema de recaudación de patentes.**

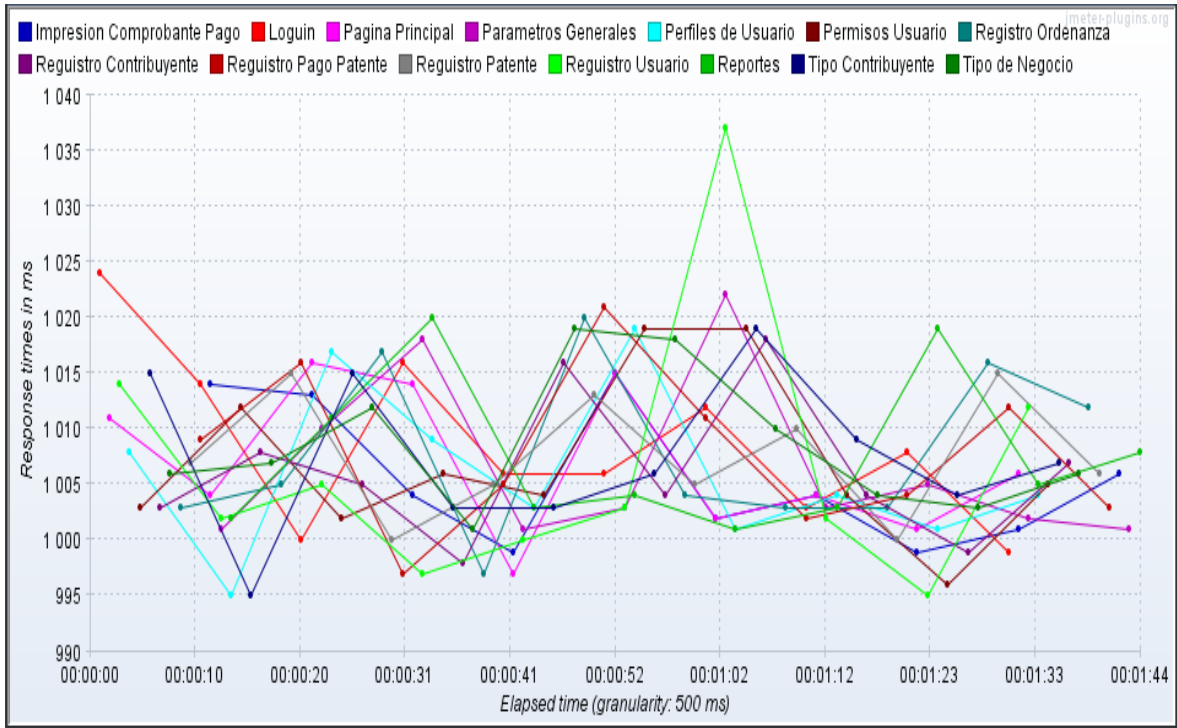
Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	16:08:25.657	Pruebas Gad Penipe ...	Loguin	1005		1964
2	16:08:26.663	Pruebas Gad Penipe ...	Loguin	1013		1964
3	16:08:26.667	Pruebas Gad Penipe ...	Pagina Principal	1009		2709
4	16:08:27.660	Pruebas Gad Penipe ...	Loguin	1015		1964
5	16:08:27.679	Pruebas Gad Penipe ...	Registro Usuario	996		2709
6	16:08:27.680	Pruebas Gad Penipe ...	Pagina Principal	1010		2709
7	16:08:28.659	Pruebas Gad Penipe ...	Loguin	1016		1964
8	16:08:28.676	Pruebas Gad Penipe ...	Perfiles de Usuario	999		2709
9	16:08:28.676	Pruebas Gad Penipe ...	Pagina Principal	1015		2709
10	16:08:28.695	Pruebas Gad Penipe ...	Registro Usuario	1002		2709
11	16:08:29.660	Pruebas Gad Penipe ...	Loguin	1014		1964
12	16:08:29.677	Pruebas Gad Penipe ...	Pagina Principal	1013		2709
13	16:08:29.677	Pruebas Gad Penipe ...	Permisos Usuario	1013		2709
14	16:08:29.708	Pruebas Gad Penipe ...	Perfiles de Usuario	997		2709
15	16:08:29.708	Pruebas Gad Penipe ...	Registro Usuario	1002		2709
16	16:08:30.643	Pruebas Gad Penipe ...	Loguin	1016		1964
17	16:08:30.676	Pruebas Gad Penipe ...	Pagina Principal	1014		2709
18	16:08:30.698	Pruebas Gad Penipe ...	Registro Usuario	1007		2709
19	16:08:30.698	Pruebas Gad Penipe ...	Tipo Contribuyente	1007		2709
20	16:08:30.711	Pruebas Gad Penipe ...	Perfiles de Usuario	1010		2709
21	16:08:30.707	Pruebas Gad Penipe ...	Permisos Usuario	1014		2709
22	16:08:31.643	Pruebas Gad Penipe ...	Loguin	1016		1964
23	16:08:31.660	Pruebas Gad Penipe ...	Pagina Principal	999		2709
24	16:08:31.691	Pruebas Gad Penipe ...	Registro Usuario	999		2709
25	16:08:31.706	Pruebas Gad Penipe ...	Registro Contribuyente	1000		2709
26	16:08:31.722	Pruebas Gad Penipe ...	Permisos Usuario	1000		2709
27	16:08:31.706	Pruebas Gad Penipe ...	Perfiles de Usuario	1016		2709
28	16:08:31.722	Pruebas Gad Penipe ...	Tipo Contribuyente	1000		2709
29	16:08:32.644	Pruebas Gad Penipe ...	Loguin	1016		1964
30	16:08:32.661	Pruebas Gad Penipe ...	Pagina Principal	1014		2709
31	16:08:32.660	Pruebas Gad Penipe ...	Registro Usuario	1015		2709
32	16:08:32.699	Pruebas Gad Penipe ...	Perfiles de Usuario	1007		2709
33	16:08:32.706	Pruebas Gad Penipe ...	Tipo de Negocio	1000		2709
34	16:08:32.723	Pruebas Gad Penipe ...	Registro Contribuyente	1015		2709
35	16:08:32.723	Pruebas Gad Penipe ...	Tipo Contribuyente	1015		2709
36	16:08:32.723	Pruebas Gad Penipe ...	Permisos Usuario	1015		2709
37	16:08:33.644	Pruebas Gad Penipe ...	Loguin	1015		1964
38	16:08:33.661	Pruebas Gad Penipe ...	Pagina Principal	1014		2709
39	16:08:33.690	Pruebas Gad Penipe ...	Registro Usuario	1001		2709
40	16:08:33.690	Pruebas Gad Penipe ...	Perfiles de Usuario	1016		2709
41	16:08:33.707	Pruebas Gad Penipe ...	Registro Ordenanza	999		2709
42	16:08:33.707	Pruebas Gad Penipe ...	Permisos Usuario	1015		2709
43	16:08:33.741	Pruebas Gad Penipe ...	Tipo Contribuyente	996		2709
44	16:08:33.741	Pruebas Gad Penipe ...	Registro Contribuyente	1012		2709
45	16:08:33.741	Pruebas Gad Penipe ...	Tipo de Negocio	1012		2709
46	16:08:34.644	Pruebas Gad Penipe ...	Loguin	1011		1964
47	16:08:34.661	Pruebas Gad Penipe ...	Pagina Principal	994		2709



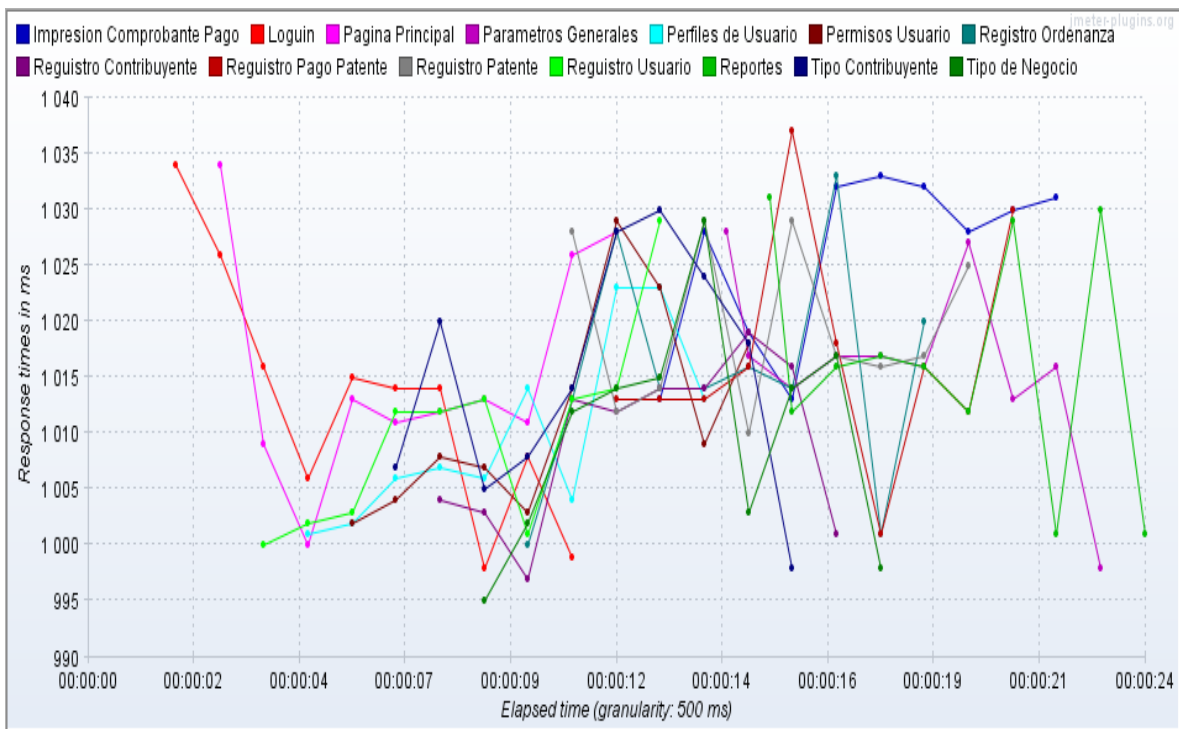
48	16:08:34.692	Pruebas Gad Penipe ...	Perfiles de Usuario	995		2709
49	16:08:34.689	Pruebas Gad Penipe ...	Registro Usuario	1013		2709
50	16:08:34.707	Pruebas Gad Penipe ...	Permisos Usuario	1011		2709
51	16:08:34.708	Pruebas Gad Penipe ...	Registro Patente	1010		2709
52	16:08:34.739	Pruebas Gad Penipe ...	Registro Contribuyente	995		2709
53	16:08:34.723	Pruebas Gad Penipe ...	Tipo Contribuyente	1011		2709
54	16:08:34.755	Pruebas Gad Penipe ...	Registro Ordenanza	1010		2709
55	16:08:34.755	Pruebas Gad Penipe ...	Tipo de Negocio	1010		2709
56	16:08:35.656	Pruebas Gad Penipe ...	Registro Usuario	1005		2709
57	16:08:35.657	Pruebas Gad Penipe ...	Pagina Principal	1004		2709
58	16:08:35.688	Pruebas Gad Penipe ...	Permisos Usuario	1005		2709
59	16:08:35.703	Pruebas Gad Penipe ...	Perfiles de Usuario	1006		2709
60	16:08:35.719	Pruebas Gad Penipe ...	Tipo Contribuyente	1005		2709
61	16:08:35.720	Pruebas Gad Penipe ...	Registro Pago Patente	1004		2709
62	16:08:35.735	Pruebas Gad Penipe ...	Registro Contribuyente	1005		2709
63	16:08:35.735	Pruebas Gad Penipe ...	Tipo de Negocio	1005		2709
64	16:08:35.767	Pruebas Gad Penipe ...	Registro Ordenanza	1005		2709
65	16:08:35.767	Pruebas Gad Penipe ...	Registro Patente	1005		2709
66	16:08:36.662	Pruebas Gad Penipe ...	Perfiles de Usuario	1019		2709
67	16:08:36.662	Pruebas Gad Penipe ...	Registro Usuario	1019		2709
68	16:08:36.694	Pruebas Gad Penipe ...	Tipo Contribuyente	1018		2709
69	16:08:36.725	Pruebas Gad Penipe ...	Registro Contribuyente	1002		2709
70	16:08:36.710	Pruebas Gad Penipe ...	Permisos Usuario	1017		2709
71	16:08:36.741	Pruebas Gad Penipe ...	Registro Ordenanza	1002		2709
72	16:08:36.741	Pruebas Gad Penipe ...	Tipo de Negocio	1002		2709
73	16:08:36.726	Pruebas Gad Penipe ...	Impresion Comprobante P...	1017		2709
74	16:08:36.774	Pruebas Gad Penipe ...	Registro Patente	1001		2709
75	16:08:36.774	Pruebas Gad Penipe ...	Registro Pago Patente	1001		2709
76	16:08:37.682	Pruebas Gad Penipe ...	Perfiles de Usuario	1014		2709
77	16:08:37.682	Pruebas Gad Penipe ...	Permisos Usuario	1014		2709
78	16:08:37.712	Pruebas Gad Penipe ...	Registro Contribuyente	1000		2709
79	16:08:37.728	Pruebas Gad Penipe ...	Tipo de Negocio	999		2709
80	16:08:37.728	Pruebas Gad Penipe ...	Tipo Contribuyente	1015		2709
81	16:08:37.744	Pruebas Gad Penipe ...	Registro Ordenanza	1015		2709
82	16:08:37.744	Pruebas Gad Penipe ...	Registro Patente	1015		2709
83	16:08:37.744	Pruebas Gad Penipe ...	Parametros Generales	1015		2709
84	16:08:37.777	Pruebas Gad Penipe ...	Registro Pago Patente	997		2709
85	16:08:37.777	Pruebas Gad Penipe ...	Impresion Comprobante P...	997		2709
86	16:08:38.697	Pruebas Gad Penipe ...	Tipo Contribuyente	1000		2709
87	16:08:38.698	Pruebas Gad Penipe ...	Permisos Usuario	999		2709
88	16:08:38.728	Pruebas Gad Penipe ...	Tipo de Negocio	1000		2709
89	16:08:38.728	Pruebas Gad Penipe ...	Registro Ordenanza	1000		2709
90	16:08:38.759	Pruebas Gad Penipe ...	Registro Pago Patente	1001		2709
91	16:08:38.743	Pruebas Gad Penipe ...	Registro Contribuyente	1017		2709
92	16:08:38.759	Pruebas Gad Penipe ...	Registro Patente	1001		2709
93	16:08:38.759	Pruebas Gad Penipe ...	Reportes	1001		2709
94	16:08:38.775	Pruebas Gad Penipe ...	Impresion Comprobante P...	1016		2709
95	16:08:38.775	Pruebas Gad Penipe ...	Parametros Generales	1016		2709
96	16:08:39.699	Pruebas Gad Penipe ...	Registro Contribuyente	999		2709
97	16:08:39.729	Pruebas Gad Penipe ...	Registro Patente	1000		2709
98	16:08:39.729	Pruebas Gad Penipe ...	Registro Ordenanza	1001		2709
99	16:08:39.699	Pruebas Gad Penipe ...	Tipo Contribuyente	1014		2709
100	16:08:39.761	Pruebas Gad Penipe ...	Impresion Comprobante P...	999		2709



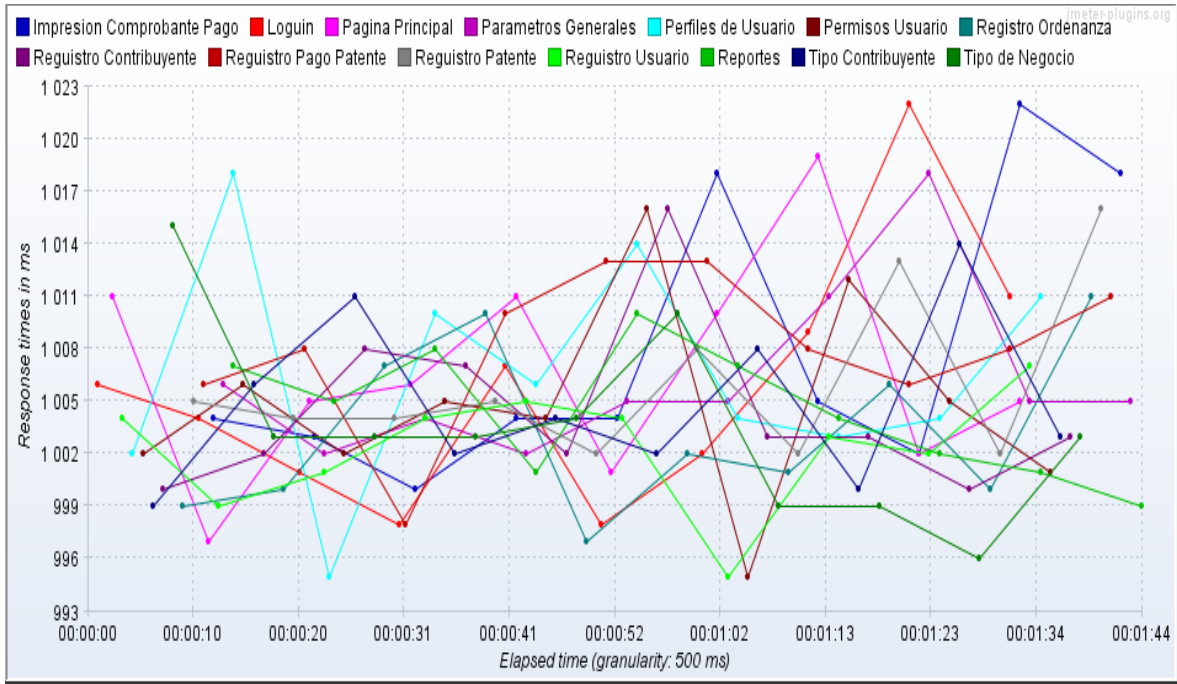
101	16:08:39.761	Pruebas Gad Penipe ...	Registro Pago Patente	999		2709
102	16:08:39.762	Pruebas Gad Penipe ...	Tipo de Negocio	1014		2709
103	16:08:39.793	Pruebas Gad Penipe ...	Parametros Generales	1014		2709
104	16:08:39.793	Pruebas Gad Penipe ...	Reportes	1014		2709
105	16:08:40.699	Pruebas Gad Penipe ...	Tipo de Negocio	1003		2709
106	16:08:40.731	Pruebas Gad Penipe ...	Registro Contribuyente	1002		2709
107	16:08:40.732	Pruebas Gad Penipe ...	Registro Pago Patente	1001		2709
108	16:08:40.731	Pruebas Gad Penipe ...	Registro Patente	1002		2709
109	16:08:40.761	Pruebas Gad Penipe ...	Impresion Comprobante P...	1003		2709
110	16:08:40.761	Pruebas Gad Penipe ...	Parametros Generales	1003		2709
111	16:08:40.776	Pruebas Gad Penipe ...	Registro Ordenanza	1004		2709
112	16:08:40.809	Pruebas Gad Penipe ...	Reportes	1002		2709
113	16:08:41.703	Pruebas Gad Penipe ...	Registro Ordenanza	1020		2709
114	16:08:41.734	Pruebas Gad Penipe ...	Impresion Comprobante P...	1020		2709
115	16:08:41.734	Pruebas Gad Penipe ...	Registro Pago Patente	1004		2709
116	16:08:41.734	Pruebas Gad Penipe ...	Tipo de Negocio	1004		2709
117	16:08:41.765	Pruebas Gad Penipe ...	Reportes	1004		2709
118	16:08:41.781	Pruebas Gad Penipe ...	Registro Patente	1004		2709
119	16:08:41.765	Pruebas Gad Penipe ...	Parametros Generales	1020		2709
120	16:08:42.724	Pruebas Gad Penipe ...	Registro Patente	1018		2709
121	16:08:42.755	Pruebas Gad Penipe ...	Impresion Comprobante P...	1018		2709
122	16:08:42.755	Pruebas Gad Penipe ...	Registro Ordenanza	1003		2709
123	16:08:42.755	Pruebas Gad Penipe ...	Parametros Generales	1003		2709
124	16:08:42.785	Pruebas Gad Penipe ...	Reportes	1004		2709
125	16:08:42.785	Pruebas Gad Penipe ...	Registro Pago Patente	1020		2709
126	16:08:43.743	Pruebas Gad Penipe ...	Registro Pago Patente	1017		2709
127	16:08:43.776	Pruebas Gad Penipe ...	Reportes	1015		2709
128	16:08:43.776	Pruebas Gad Penipe ...	Parametros Generales	1015		2709
129	16:08:43.776	Pruebas Gad Penipe ...	Registro Patente	1015		2709
130	16:08:43.806	Pruebas Gad Penipe ...	Impresion Comprobante P...	1016		2709
131	16:08:44.761	Pruebas Gad Penipe ...	Impresion Comprobante P...	1002		2709
132	16:08:44.795	Pruebas Gad Penipe ...	Reportes	1015		2709
133	16:08:44.795	Pruebas Gad Penipe ...	Registro Pago Patente	1015		2709
134	16:08:44.823	Pruebas Gad Penipe ...	Parametros Generales	1002		2709
135	16:08:45.764	Pruebas Gad Penipe ...	Parametros Generales	1018		2709
136	16:08:45.810	Pruebas Gad Penipe ...	Impresion Comprobante P...	1003		2709
137	16:08:45.826	Pruebas Gad Penipe ...	Reportes	1019		2709
138	16:08:46.783	Pruebas Gad Penipe ...	Reportes	1003		2709
139	16:08:46.814	Pruebas Gad Penipe ...	Parametros Generales	1003		2709
140	16:08:47.819	Pruebas Gad Penipe ...	Reportes	1014		2709



**Figura 20:** Medición Tiempo de respuesta.

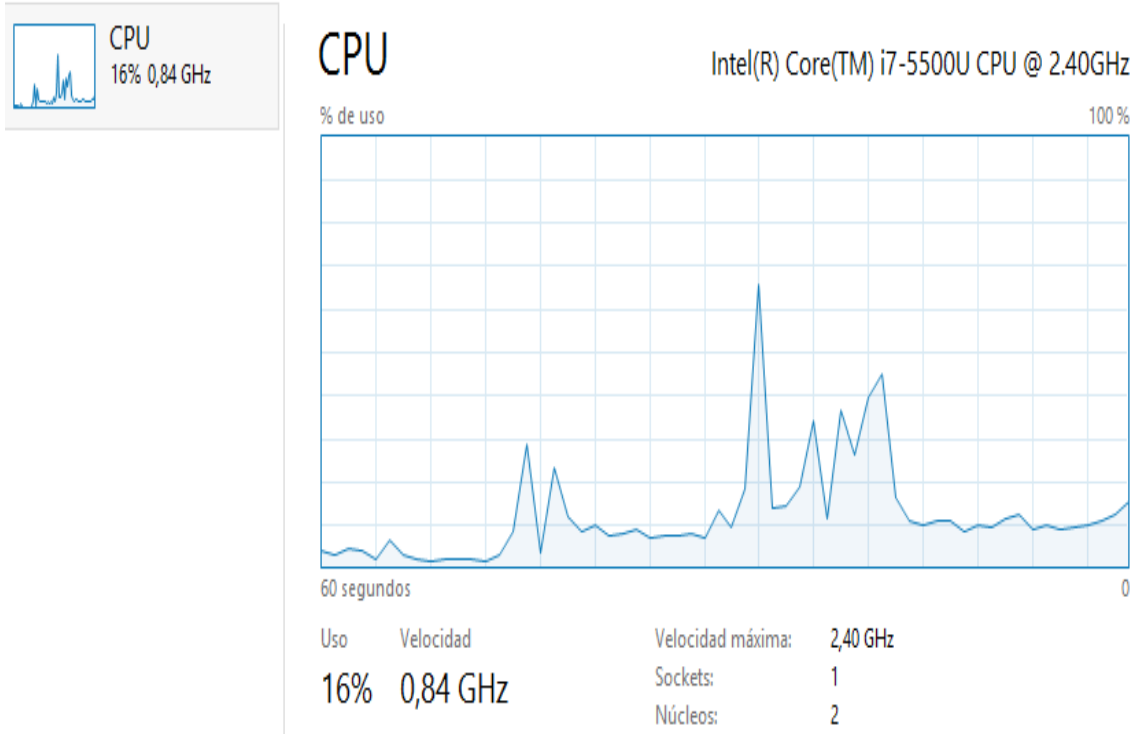


**Figura 21:** Medición Tiempo de respuesta.

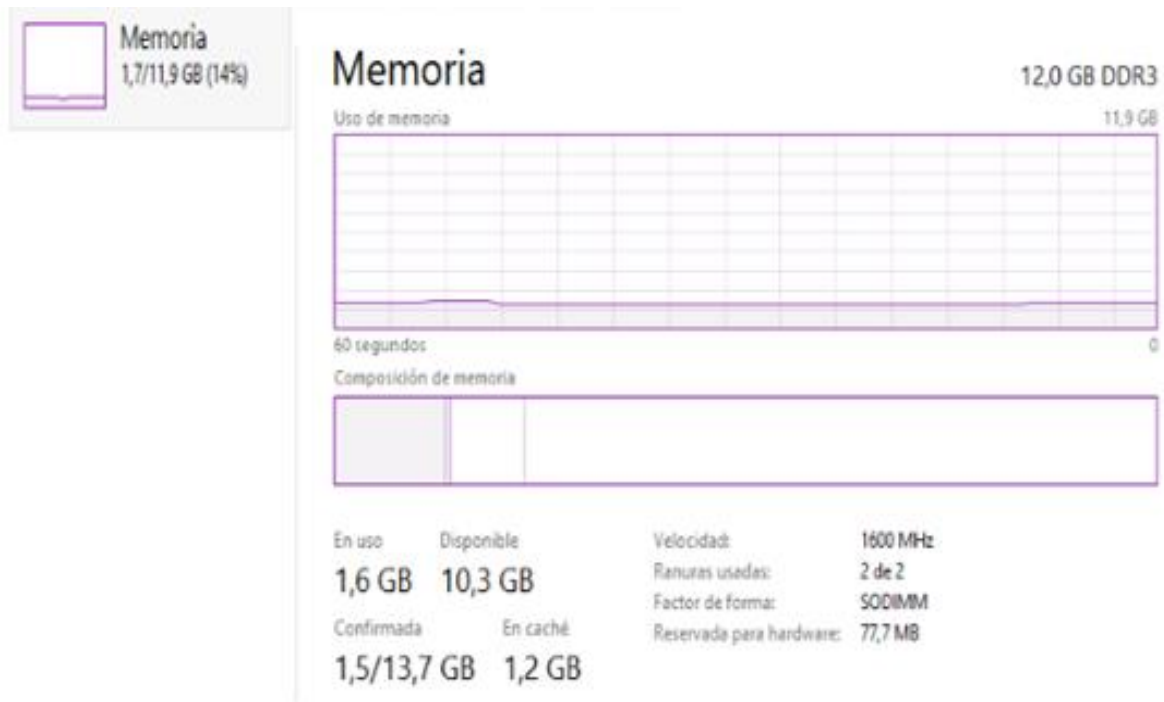


**Figura 22:** Medición Tiempo de respuesta.

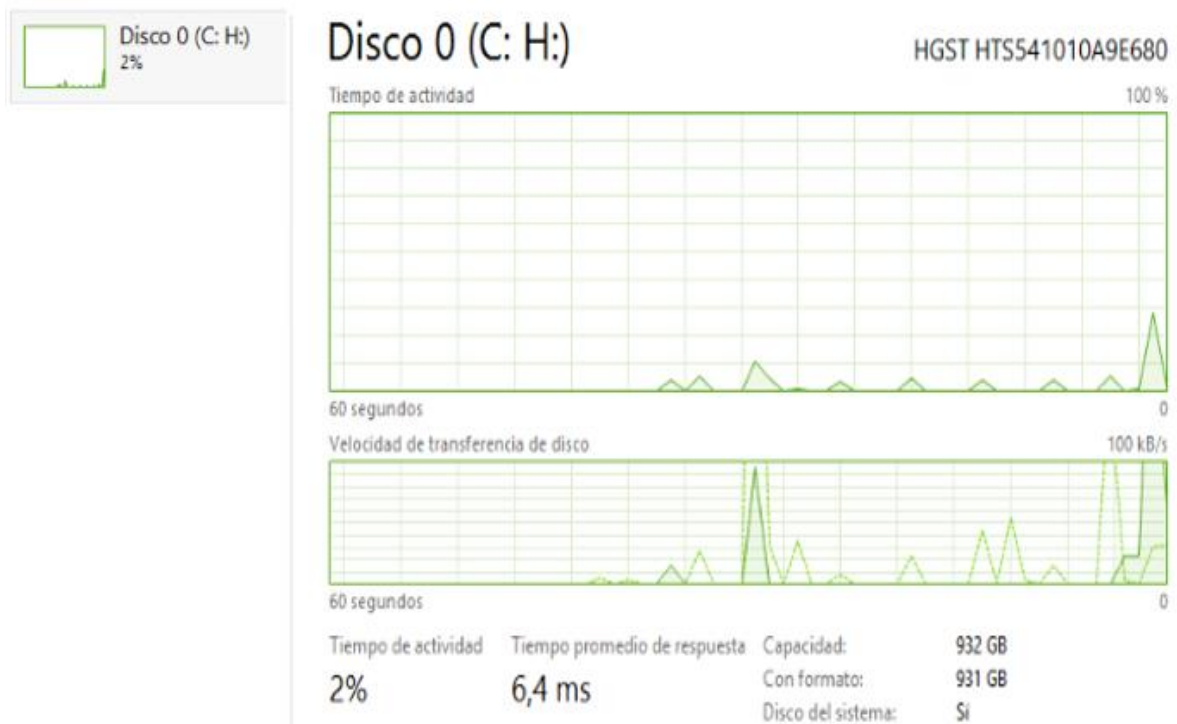
**ANEXO IV: Resultado medición del uso de recursos del sistema de recaudación de patentes.**



**Figura 23:** Uso de CPU.



**Figura 24:** Uso de memoria.



**Figura 25:** Uso de Disco.



**ANEXO V: Certificado de desarrollo.**



**CERTIFICACIÓN**

**N° 2020-0022**

A petición verbal del interesado; la suscrita Ing. Gabriela Vinueza, Jefe de Informática del GAD Municipal del Cantón Penipe, CERTIFICA:

Que, el Sr. ALEX BLADIMIR MANOBANDA USHCA, portador de la cedula de ciudadanía N° 0604413831, egresado de la Universidad Nacional de Chimborazo de la Facultad de Ingeniería de la Carrera de Sistemas y Computación, desarrolló y entregó el sistema web de Recaudación de Patentes para el GADM Penipe cumpliendo con todos los requisitos establecidos en el proyecto como parte de su tema de Tesis de Grado "MODELO FURPS PARA EVALUAR EL SISTEMA WEB DE RECAUDACION DE PATENTES GADM PENIPE".

Es todo cuanto puedo certificar en honor a la verdad, facultando al interesado hacer uso del presente para los fines que crea conveniente.

Penipe, 28 de julio del 2020

Atentamente,

  
Ing. Gabriela Vinueza  
**JEFE DE INFORMÁTICA**



DIRECCIÓN: Silvio Hato 09-21 y David Ramos  
TELÉFONOS 032 907196 / 032 907197  
MAIL: municipio\_de\_penipe@hotmail.com

**Figura 26:** Certificado de Desarrollo del Sistema Web.