



# **UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO**



**FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
ESCUELA PROFESIONAL DE INGENIERÍA EN COMPUTACIÓN E INFORMÁTICA**

**“DESARROLLO DE UNA WEB SERVICE PARA FACILITAR EL  
ACCESO DE INFORMACIÓN DEL ÁREA DE RENTAS A LOS  
CIUDADANOS DEL DISTRITO DE JOSÉ LEONARDO ORTIZ -  
CHICLAYO – 2017”**

**PARA OPTAR EL TÍTULO PROFESIONAL DE:**

Ingeniero en Computación e Informática

**PRESENTADO POR:**

Bach. Bravo Merlo, Cristian Jonathan

Bach. Ríos Zuloeta, Edson Alberto

Lambayeque 2020

**“DESARROLLO DE UNA WEB SERVICE PARA FACILITAR EL ACCESO DE INFORMACIÓN DEL ÁREA DE RENTAS A LOS CIUDADANOS DEL DISTRITO DE JOSÉ LEONARDO ORTIZ – CHICLAYO 2017”.**

**PRESENTADO POR:**

---

**Bach. BRAVO MERLO, Cristian Jonathan**

---

**Bach. RÍOS ZULOETA, Edson Alberto**

**ASESORADO POR:**

---

**Dra. Ing. LECCA ORREGO, Giuliana Fiorella**

**APROBADO POR:**

---

**Dr. Ing. MORENO HEREDIA, Armando José**

---

**Dra. Ing. BRAVO JAICO, Jessie**

---

**Ing. TERÁN SANTA CRUZ, Franklin**

## **DEDICATORIA**

### **A Dios**

Por habernos permitido llegar hasta este punto y dado salud para lograr nuestros objetivos, por los triunfos y los momentos difíciles que nos han enseñado a valorar cada día más, además por su infinita bondad y amor.

### **A nuestros padres**

Por habernos apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que nos ha permitido ser unas personas de bien, pero más que nada por su amor.

Por los ejemplos de perseverancia y constancia que los caracterizan y que nos ha infundado siempre, por el valor mostrado para salir adelante y por su amor.

Le agradecemos a cada una de las personas que nos supo apoyar en cada paso de la elaboración de esta tesis.

## **RESUMEN**

Una web service es un conjunto de protocolos y estándares que nos permite la comunicación entre programas que han sido creados en distintos lenguajes. En la actualidad el uso de los webs service para Android se hace cada vez más común, ya que con más frecuencia necesitamos la comunicación y transferencia de datos entre nuestro dispositivo móvil y los grandes administradores de base de datos como son: SQL, Oracle, MySQL, entre otros.

Este proyecto de investigación tiene como objetivo desarrollar una web service que permitirá al usuario optimizar el tiempo en las operaciones a realizar, logrando satisfacer a los ciudadanos del distrito de José Leonardo Ortiz - Chiclayo. Se constató que la Municipalidad Distrital de José Leonardo Ortiz lidia con el problema de ofrecer una mejor calidad de servicios en aspectos relacionados a costos y tiempo de atención; como lo es: la aglomeración de clientes para solicitar información lo que a su vez genera saturación en el personal de atención e incrementa los costos de impresión, los horarios de atención, la saturación de las líneas telefónicas para consultas y la movilización del ciudadano hacia la Municipalidad Distrital de José Leonardo Ortiz.

El desarrollo de la web service para Android en PHP mejora la calidad de servicio a sus clientes de la Municipalidad de José Leonardo Ortiz en aspectos relacionados a costos y tiempo de atención, en la que el cliente puede acceder directamente a la información relacionada con sus deberes (rentas, impuestos, etc.) las 24 horas del día y los 7 días de la semana, así mismo, puede visualizar la información de sus tributos desde cualquier parte del mundo que se encuentre el cliente.

.

## **ABSTRACT**

A web service is a set of standards and protocols that allows communication between programs that have been created in different languages. At present the use of the web service for Android becomes increasingly common, since most frequently we need communication and data transfer between your mobile device and the large administrators of database such as: SQL, Oracle, MySQL, among others.

This research project aims to develop a web service that will allow the user to optimize the time in the operations to be carried out, satisfying the citizens of the district of José Leonardo Ortiz - Chiclayo. It was found that the District Municipality of José Leonardo Ortiz deals with the problem of offering a better quality of services in aspects related to costs and attention time; as it is: the agglomeration of clients to request information which in turn generates saturation in the attention staff and increases the printing costs, the opening hours, the saturation of the telephone lines for consultations and the mobilization of the citizen towards the District Municipality of José Leonardo Ortiz.

The development of the web service for Android in PHP improves the quality of service to its clients of the Municipality of José Leonardo Ortiz in aspects related to costs and time of attention, in which the client can directly access the information related to his duties (income, taxes, etc.) 24 hours a day and 7 days a week, likewise, you can view your tax information from anywhere in the world that the customer is.

## ÍNDICE GENERAL

DEDICATORIA.....	iii
RESUMEN.....	iv
ABSTRACT .....	v
ÍNDICE GENERAL.....	vi
ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	ix
ÍNDICE DE CAPTURAS DE PANTALLA .....	ix
INTRODUCCIÓN .....	11
CAPÍTULO I.....	13
I. DATOS GENERALES DE LA ORGANIZACIÓN .....	14
1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN.....	14
1.2. MISIÓN Y VISIÓN DE LA ORGANIZACIÓN .....	14
1.2.1. MISIÓN.....	14
1.2.2. VISIÓN .....	14
1.3. ORGANIGRAMA DE LA ORGANIZACIÓN.....	15
CAPÍTULO II.....	16
II. PROBLEMÁTICA DE LA INVESTIGACIÓN .....	17
2.1. REALIDAD PROBLEMÁTICA .....	17
2.1.1. PLANTEAMIENTO DEL PROBLEMA .....	17
2.2. FORMULACIÓN DEL PROBLEMA .....	18
2.3. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN .....	18
2.3.1. JUSTIFICACIÓN .....	18
2.3.2. IMPORTANCIA.....	19
2.4. OBJETIVOS DE LA INVESTIGACIÓN .....	19
2.4.1. OBJETIVO GENERAL .....	19
2.4.2. OBJETIVOS ESPECÍFICOS .....	19
CAPÍTULO III.....	20
III. MARCO METODOLÓGICO.....	21
3.1. TIPO DE INVESTIGACIÓN.....	21
3.2. HIPÓTESIS.....	21
3.3. VARIABLES.....	21
3.3.1. VARIABLE INDEPENDIENTE.....	21
3.3.2. VARIABLE DEPENDIENTE .....	21

CAPÍTULO IV.....	22
IV. MARCO TEÓRICO .....	23
4.1. ANTECEDENTES DE LA INVESTIGACIÓN .....	23
4.1.1. ANTECEDENTES EN EL CONTEXTO INTERNACIONAL.....	23
4.1.2. ANTECEDENTES EN EL CONTEXTO NACIONAL.....	24
4.1.3. ANTECEDENTES EN EL CONTEXTO LOCAL .....	25
4.2. BASE TEÓRICA .....	26
4.2.1. ANÁLISIS.....	26
4.2.1.1. SCRUM.....	26
4.2.2. HERRAMIENTAS DE DESARROLLO .....	32
4.2.2.1. XAMPP .....	32
4.2.2.2. SQL SERVER 2012 .....	33
4.2.2.3. PHP (PHP Versión 7.2.6).....	34
4.2.2.4. POSTMAN .....	35
4.2.2.5. VISUAL STUDIO CODE.....	36
4.2.2.6. FRAMEWORK CODEIGNITER 3.9 .....	38
4.2.2.7. ANDROID STUDIO (API 27) .....	39
4.2.2.8. MICROSOFT AZURE .....	42
4.2.3. TECNOLOGÍA DE DESARROLLO DE LA WEB SERVICE.....	43
4.2.3.1. JWT TOKEN .....	43
4.2.3.2. VOLLEY ANDROID .....	46
4.2.3.3. JSON .....	47
4.2.3.4. REST_CONTROLLER .....	49
CAPÍTULO V.....	50
V. DESARROLLO DE LA PROPUESTA.....	51
5.1. PLANTEAMIENTO DEL PROYECTO.....	51
5.1.1. IDENTIFICACIÓN DEL ESCENARIO DEL PROYECTO.....	51
5.1.2. IDENTIFICACIÓN DE LOS USUARIOS Y ÁREAS INTERVENIENTES .....	52
5.1.2.1. USUARIOS .....	52
5.1.3. ANÁLISIS DE FACTIBILIDAD .....	52
5.1.3.1. FACTIBILIDAD OPERATIVA.....	52
5.1.3.2. FACTIBILIDAD TÉCNICA.....	52
5.2. DEFINICIÓN DE REQUERIMIENTOS DEL NEGOCIO.....	53



5.2.1.	ANÁLISIS DEL REQUERIMIENTO.....	53
5.2.2.	REQUERIMIENTOS FUNCIONALES .....	53
5.2.2.1.	IMPLEMENTACIÓN DE LA WEB SERVICES .....	53
5.2.2.2.	DESARROLLO DEL SISTEMA WEB (Tesis_Muni).....	53
5.2.2.3.	DESARROLLO DEL SISTEMA WEB (Tesis_Banco) .....	53
5.2.2.4.	CONSULTA DE INFORMACIÓN.....	53
5.2.3.	REQUERIMIENTOS NO FUNCIONALES .....	53
5.2.3.1.	APARIENCIA O INTERFAZ.....	53
5.2.3.2.	USABILIDAD .....	54
5.2.3.3.	RENDIMIENTO .....	54
5.2.3.4.	SOPORTE Y PORTABILIDAD.....	54
5.3.	DISEÑO Y DESARROLLO DE PRESENTACIÓN DE DATOS.....	54
5.3.1.	DISEÑO ARQUITECTÓNICO .....	55
5.3.2.	PETICIONES GET.....	58
5.3.3.	PETICIONES POST.....	60
5.3.4.	DIAGRAMA DE LA BASE DE DATOS CON LAS TABLAS INVOLUCRADAS....	62
5.3.5.	CONFIGURACIÓN DEL SERVIDOR AZURE.....	63
5.3.6.	CONFIGURACIÓN DE HERRAMIENTAS Y DRIVERS .....	69
5.3.7.	CONFIGURACIÓN DE USUARIOS DE SQL SERVER .....	71
5.3.8.	CONFIGURACIÓN DE LA WEB SERVICES .....	71
5.3.9.	INTRODUCCIÓN AL ACCESO DEL SERVICIO .....	77
5.3.9.1.	ENVIAR COBROS FILTRADO .....	78
5.3.9.2.	ENVIAR COBROS GLOBAL .....	79
5.3.9.3.	SUBIR ARCHIVO .....	80
CAPÍTULO VI.....		89
VI.	ANÁLISIS DE COSTOS .....	90
6.1.	COSTO DE PERSONAL.....	90
6.2.	COSTO DE SERVICIOS .....	90
6.3.	COSTO DE HARDWARE.....	90
6.4.	COSTO DE SOFTWARE.....	90
6.5.	RESUMEN DE COSTOS DE INVERSIÓN .....	91
6.6.	VIABILIDAD DEL PROYECTO .....	91
CAPÍTULO VII.....		93

VII.	CONCLUSIONES Y RECOMENDACIONES .....	94
	REFERENCIAS BIBLIOGRÁFICAS.....	96

## ÍNDICE DE TABLAS

Tabla 1: Variable independiente .....	21
Tabla 2: Variable dependiente .....	21
Tabla 3: SCRUM vs. RUP Fuente: Elaboración propia .....	31
Tabla 4: Formato de un JWT Autor: Rodrigo Paszniuk .....	44
Tabla 5: Contenido Header Autor: Rodrigo Paszniuk .....	44
Tabla 6: Contenido Payload I Autor: Rodrigo Paszniuk .....	45
Tabla 7: Contenido Payload II Autor: Rodrigo Paszniuk .....	45
Tabla 8: Contenido Signature Autor: Rodrigo Paszniuk .....	46
Tabla 9: Contenido JSON Web Token Autor: Rodrigo Paszniuk .....	46
Tabla 10: Peticiones POST / GET Municipalidad (Android) Autor: Relaboración propia .....	56
Tabla 11: Peticiones POST / GET Municipalidad (Web) Autor: Elaboración propia .....	57
Tabla 12: Peticiones POST / GET Banco (Web) Autor: Elaboración propia .....	57
Tabla 13: Costo de Personal Autor: Elaboración propia .....	90
Tabla 14: Costo de Servicios Autor: Elaboración propia .....	90
Tabla 15: Costo de Hardware Autor: Elaboración propia .....	90
Tabla 16: Costo de Software Autor: Elaboración propia .....	90
Tabla 17: Resumen de costos de inversión Autor: Elaboración propia .....	91

## ÍNDICE DE FIGURAS

Figura 1: Organigrama de la Municipalidad Distrital de José Leonardo Ortiz .....	15
Figura 2: Evolución histórica de la gestión de proyectos Autor: Manuel Trigas Gallegos .....	28
Figura 3: Elementos básicos que formarán una fase Autor: Manuel Trigas Gallegos .....	30
Figura 4: Requerimientos del Sistema Autor: Academia Android .....	40
Figura 5: Comparativa entre Android Studio y ADT Eclipse Autor: Academia Android .....	41
Figura 6: Municipalidad Distrital de José Leonardo Ortiz .....	51
Figura 7: Diseño arquitectónico Autor: Elaboración propia: .....	55
Figura 8: Petición GET Autor: Elaboración propia .....	59
Figura 9: Petición POST Autor: Elaboración propia .....	61

## ÍNDICE DE CAPTURAS DE PANTALLA

Captura de pantalla 1: Creación del servidor y de la base de datos .....	63
Captura de pantalla 2: Información de la base de datos y características .....	63
Captura de pantalla 3: Configuración de firewall .....	64
Captura de pantalla 4: Agregando IP del cliente .....	64
Captura de pantalla 5: Cadena de conexión .....	65
Captura de pantalla 6: Elegir dirección IP pública .....	65
Captura de pantalla 7: Creación dirección IP pública .....	66

Captura de pantalla 8: Elegir y crear grupo de seguridad de red .....	66
Captura de pantalla 9: Máquina virtual y restauración de la base de datos .....	67
Captura de pantalla 10: Iniciación de restauración de la base de datos .....	67
Captura de pantalla 11: Finalización de restauración de la base de datos .....	68
Captura de pantalla 12: Agregando los archivos .dll para el funcionamiento SQL Server .....	69
Captura de pantalla 13: Abriendo carpeta para la configuración de las extensiones del driver SQL Server .....	70
Captura de pantalla 14: Configurando las extensiones SQL Server del xampp .....	70
Captura de pantalla 15: Iniciando sesión de la base de datos SQL Server .....	71
Captura de pantalla 16: Configuración de la conexión a la base de datos de la Web Services ..	71
Captura de pantalla 17: Configuración del tiempo de expiración de los usuarios del sistema, claves secretas .....	72
Captura de pantalla 18: Configuración de la ruta base para acceder a la web services .....	72
Captura de pantalla 19: Configuración de las rutas de accesos a la web services con las peticiones HTTP .....	73
Captura de pantalla 20: Agregando la librería JWT y REST_Controller .....	73
Captura de pantalla 21: Configurando la librería REST_Controller a nuestro controlador Auth	74
Captura de pantalla 22: Demostración de nuestra librería JWT, salida de datos .....	74
Captura de pantalla 23: Realización de pruebas de datos de nuestra web services haciendo uso de POSTMAN .....	75
Captura de pantalla 24: Cargando la librería Volley y Gson para nuestra aplicación móvil .....	75
Captura de pantalla 25: Configuración de la librería Volley para la conexión a nuestra web services .....	76
Captura de pantalla 26: Muestra del código de la autenticación de nuestra aplicación móvil ..	76
Captura de pantalla 27: Inicio de sesión .....	77
Captura de pantalla 28: Panel de control principal .....	77
Captura de pantalla 29: Búsqueda de deudas por mes y año .....	78
Captura de pantalla 30: Envío de las deudas de los contribuyentes al banco .....	79
Captura de pantalla 31: Listado de deudas de los contribuyentes .....	79
Captura de pantalla 32: Inicio de sesión BancoTesis .....	82
Captura de pantalla 33: Listado de los contribuyentes deudores BancoTesis .....	82
Captura de pantalla 34: Realizar el pago de un contribuyente .....	83
Captura de pantalla 35: Pago realizado de un impuesto del contribuyente .....	84
Captura de pantalla 36: Listado de pagos para enviar a la Municipalidad de José Leonardo Ortiz .....	85
Captura de pantalla 37: Inicio de Sesión .....	86
Captura de pantalla 38: Perfil de Usuario .....	86
Captura de pantalla 39: Menú de Opciones .....	87
Captura de pantalla 40: Vista de Estado de Predio .....	87
Captura de pantalla 41: Cuotas por Año .....	87
Captura de pantalla 42: Resumen por Año .....	87
Captura de pantalla 43: Vista de Estado Arbitrios .....	88
Captura de pantalla 44: Vista de Pagos .....	88
Captura de pantalla 45: Web de la Municipalidad de José Leonardo Ortiz .....	88

## INTRODUCCIÓN

Dentro de la Municipalidad Distrital de José Leonardo Ortiz se está construyendo la Alianza para el Gobierno Abierto que es una iniciativa internacional voluntaria y multisectorial que pretende asegurar compromisos concretos de los gobiernos a sus ciudadanos para promover la transparencia, empoderar a los ciudadanos, luchar contra la corrupción y aprovechar las nuevas tecnologías para fortalecer la gobernabilidad.

Una Institución que abre sus puertas y nos invita a participar, nos escucha y dialoga y reconoce que las buenas ideas vienen de todas partes.

Una Institución que brinda información y abre sus datos para el escrutinio público, una institución que rinde cuentas, que fomenta la innovación para contribuir a mejorar la educación, fomentar nuevas actividades económicas y oportunidades para todos, ¿Y cómo lograrlo? con una formula precisa, que es teniendo acceso a la información y participación como derecho, transparencia como política, datos abiertos como instrumento, gobierno legitimado, honesto, eficiente y eficaz.

A través de esta iniciativa sobre el acceso a la información a los ciudadanos es que la institución está abriendo muchas posibilidades de expansión y confiabilidad en las comunicaciones, pero nuevamente nació la misma preocupación.

La seguridad de este servicio, y es aquí en donde este trabajo de investigación busca encontrar una variación de los métodos existentes de seguridad aplicándolos a los Web Service, algunos métodos de seguridad hablan de autenticación, autorización, integridad de datos, confidencialidad, no repudiación, etc.

La creación de una web service para Android en PHP en la Municipalidad de José Leonardo Ortiz tiene como objetivo principal desarrollar una web service que permitirá al usuario optimizar el tiempo en las operaciones a realizar, logrando satisfacer a los ciudadanos del distrito de José Leonardo Ortiz – Chiclayo y como objetivos específicos: realizar un análisis detallado de los procesos del área de rentas de la Municipalidad Distrital de José Leonardo Ortiz – Chiclayo, diseñar las interfaces para la aplicación, utilizar proceso ágiles y herramientas de software para llevar a cabo el desarrollo del sistema, diseñar una solución segura que permita proteger la información personal de los clientes y definir la estructura de costos que implicará la realización de la propuesta planteada.

El usuario podrá descargar la aplicación móvil en su dispositivo con Sistema Operativo Android, lo cual podrá tener contacto con la Municipalidad las 24 horas del día, tendrá siempre la posibilidad de consultar y/o visualizar la información de sus tributos, estados de sus pagos y deudas desde cualquier lugar donde se encuentre desde la aplicación descargada que estará disponible y visible en la pantalla de su dispositivo de manera constante.

Luego de analizar estos requerimientos se podrán ver métodos de comunicación existentes, analizando sus fallos en el área de seguridad a través del estudio de los ataques más comunes existentes para los Web Service, con el fin de dar una solución final que también cubra la comunicación entre las diferentes entidades y posea la suficiente seguridad para brindar un mejor servicio.

Las investigaciones encontradas y estudiadas acerca de la seguridad de los Web Service han generado ciertos parámetros que se deben tener en cuenta al momento de implementar estos servicios para poder obtener un ámbito seguro de comunicación. Y es aquí donde en este trabajo de investigación ofrece posibilidades de expansión y de implementación de nuevas opciones de seguridad, refiriéndose a la autenticación por medio de roles para Web Service; una aplicación que, al momento de integrarla con los servicios existentes, ofrecerá una nueva visión de seguridad y le dará al Web Service una nueva dimensión en la cual explorar nuevas ideas e innovaciones.

En el Capítulo I se menciona los datos generales de la Municipalidad Distrital de José Leonardo Ortiz, para posteriormente en el Capítulo II hablar sobre la problemática en la que se basa la presente investigación, luego en el Capítulo III se detalla el marco metodológico, hipótesis y variables. En el Capítulo IV se hace una recopilación de todo el marco teórico necesario para la investigación, y así en el Capítulo V desarrollar la propuesta en base a la metodología elegida para finalmente en el Capítulo VI evaluar los resultados, costos y beneficios de la investigación realizada

**CAPÍTULO I**  
**DATOS GENERALES DE LA**  
**ORGANIZACIÓN**

## I. DATOS GENERALES DE LA ORGANIZACIÓN

### 1.1. DESCRIPCIÓN DE LA ORGANIZACIÓN

<b>Razón social</b>	: Municipalidad Distrital de José Leonardo Ortiz
<b>Giro del negocio</b>	: Sector Público
<b>Dirección</b>	: Av. Saenz Peña 2151
<b>Teléfono</b>	: 074 – 626357
<b>Página web</b>	: <a href="http://www.munijlo.gob.pe/web">www.munijlo.gob.pe/web</a>

### 1.2. MISIÓN Y VISIÓN DE LA ORGANIZACIÓN

#### 1.2.1. MISIÓN

“Mejorar la calidad de vida de la población Leonardina, dispuestos a la prestación de servicios públicos en todos los niveles, mediante la ejecución de obras de corto y largo plazo que garanticen ser oportunos y óptimos con una administración transparente, capacidad de gestión adecuada y práctica concertadora, abierta al dialogo y al cambio social.”

#### 1.2.2. VISIÓN

“Ser competentes en el Gobierno Municipal, demostrando acercamiento a los pueblos jóvenes, garantizando su bienestar y progreso acorde con la tecnología, la democracia y la ciencia.”

### 1.3. ORGANIGRAMA DE LA ORGANIZACIÓN

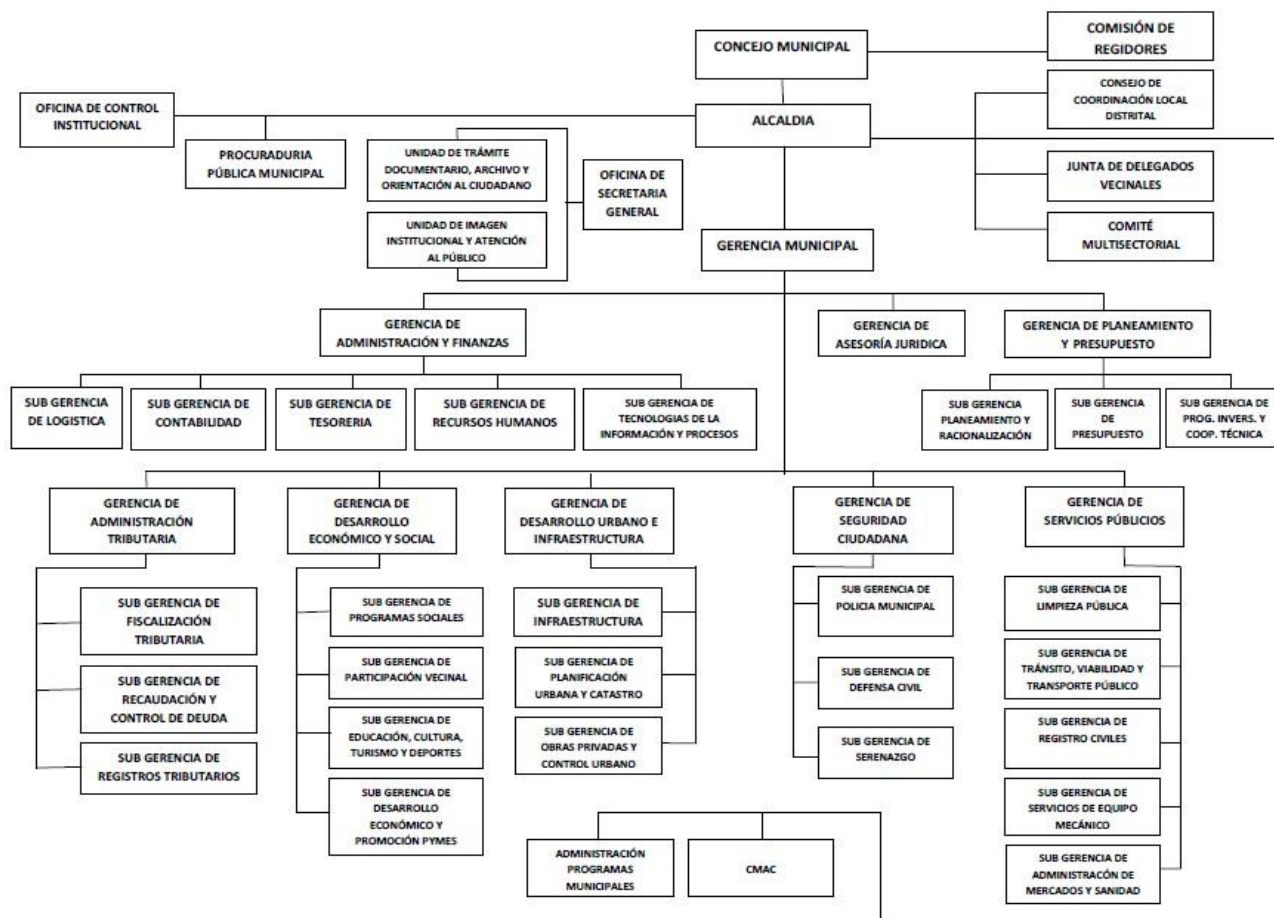


Figura 1: Organigrama de la Municipalidad Distrital de José Leonardo Ortiz

Fuente: [http://www.munijlo.gob.pe/web/archives\\_load/ORGANIGRAMA.pdf](http://www.munijlo.gob.pe/web/archives_load/ORGANIGRAMA.pdf)



## **CAPÍTULO II**

# **PROBLEMÁTICA DE LA INVESTIGACIÓN**

## **II. PROBLEMÁTICA DE LA INVESTIGACIÓN**

### **2.1. REALIDAD PROBLEMÁTICA**

#### **2.1.1. PLANTEAMIENTO DEL PROBLEMA**

El Distrito de José Leonardo Ortiz es uno de los veinte distritos de la provincia de Chiclayo, ubicada en la región Lambayeque, bajo la administración del Gobierno Regional, en el norte de Perú. Este distrito surge debido a la explotación demográfica de la ciudad de Chiclayo durante la década de 1940.

Fue creado en el segundo gobierno del presidente Manuel Prado Ugarteche, por Ley 13734 del 28 de noviembre de 1961, con el nombre de San Carlos, el cual fue cambiado por el actual el 5 de febrero de 1966.

La municipalidad distrital de José Leonardo Ortiz, como institución de gobierno local, tiene como función, la de administrar los ingresos económicos y desarrollar labores en beneficio y progreso de la comunidad local con la misión de “Mejorar la calidad de vida de la población Leonardina, dispuestos a la prestación de servicios públicos en todos los niveles, mediante la ejecución de obras de corto y largo plazo que garanticen ser oportunos y óptimos con una administración transparente, capacidad de gestión adecuada y práctica concertadora, abierta al dialogo y al cambio social” y con la visión de “Ser competentes en el Gobierno Municipal, demostrando acercamiento a los pueblos jóvenes, garantizando su bienestar y progreso acorde con la tecnología, la democracia y la ciencia”.

No hay duda que internet ha revolucionado la cultura y, por lo tanto, el funcionamiento de la sociedad actual. Numerosos países vienen realizando esfuerzos por formular declaraciones políticas en torno al aprovechamiento de las Tecnologías de la Información y las Comunicaciones (TIC), en las cuales se han incorporado conceptos asociados al Gobierno Abierto (open government) para promover una administración más eficaz, eficiente y transparente, mediante el ofrecimiento en líneas de información al ciudadano, mecanismos de control de gestión, servicios, trámites, participación en la elaboración de políticas estatales, de rendición de cuentas y responsabilización por el uso de los recursos públicos, de evaluación y control ciudadanos de los resultados gubernamentales.

La municipalidad lidia con el problema de ofrecer una mejor calidad de servicio a sus clientes en aspectos relacionados a costos y tiempos de atención. La municipalidad ha identificado estos problemas como lo es: la aglomeración de clientes para solicitar información lo que a su vez genera saturación en el personal de atención e incrementa los costos de impresión, los horarios de atención, la saturación de las líneas telefónicas para consultas y la movilización del ciudadano hacia la Municipalidad Distrital de José Leonardo Ortiz.

Según las conclusiones obtenidas de reuniones con el personal del área de rentas de la municipalidad, uno de los problemas más resaltantes es el de horarios de atención en la municipalidad. La municipalidad establece un

horario de atención fijo, pero no dispone de una atención las 24 horas del día los 7 días de la semana, lo que afecta directamente a sus cobranzas netas, ya que el disponer de esta cobertura de atención le generaría ingresos más estables.

Otro problema actual de la municipalidad es que sus clientes no pueden visualizar la información de sus tributos desde cualquier lugar donde se encuentren. El único lugar donde pueden preguntar por los mismos es en la misma municipalidad.

Por lo que proponemos desarrollar una web service para Android en PHP, en la que el cliente pueda interactuar directamente.

## 2.2. FORMULACIÓN DEL PROBLEMA

¿El desarrollo de una web service facilitará el acceso de información del área de rentas a los ciudadanos del Distrito de José Leonardo Ortiz - Chiclayo?

## 2.3. JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN

### 2.3.1. JUSTIFICACIÓN

La municipalidad lidia con el problema de ofrecer una mejor calidad de servicio a sus clientes en aspectos relacionados a costos y tiempos de atención. Por lo que proponemos desarrollar una web service para Android en PHP, en la que el cliente pueda interactuar directamente con un sistema de información. Para lo cual damos las siguientes soluciones:

- ✚ Para el problema de los horarios de atención utilizaremos tecnologías de información ya existentes que nos permiten garantizar la continuidad del negocio sin necesidad que se cuente con un personal fijo de atención y que esté limitado a un horario establecido. Estas tecnologías presentan una disposición de 24x7. Por otro lado, al momento de realizar el pago, existen sistemas de información independientes brindados por entidades bancarias, los cuales se encargan de proveer este servicio de pago electrónico. Con esto, la seguridad está garantizada de antemano debido a la credibilidad de las mismas entidades. Estos sistemas de pago electrónico son denominados “Servicio de recaudación”, y entre los más comerciales destacan: BBVA, BCP e Interbank.
- ✚ Para suplir la necesidad de que los clientes no pueden visualizar la información relacionada con sus deberes (pagos, impuestos, etc.), ya que con más frecuencia se necesita la comunicación y transferencia de datos entre nuestro dispositivo móvil y los grandes administradores de base de

datos, desde cualquier lugar donde se encuentren existen diversos aparatos electrónicos con sistema operativo Android en el cual puedan acceder.

### **2.3.2. IMPORTANCIA**

Con el desarrollo de este proyecto de investigación los ciudadanos del Distrito de José Leonardo Ortiz se pueden favorecer de manera directa con la información relacionada con sus tributos, puesto que la aplicación permitirá contar con información actualizada, confiable y organizada de tal forma que el acceso se realice en forma eficiente y fácil manejo.

## **2.4. OBJETIVOS DE LA INVESTIGACIÓN**

### **2.4.1. OBJETIVO GENERAL**

Desarrollar una web service permitirá al usuario optimizar el tiempo en las operaciones a realizar, logrando satisfacer a los ciudadanos del distrito de José Leonardo Ortiz – Chiclayo.

### **2.4.2. OBJETIVOS ESPECÍFICOS**

1. Realizar un análisis detallado de los procesos del área de rentas de la Municipalidad Distrital de José Leonardo Ortiz – Chiclayo.
2. Diseñar las interfaces para la aplicación.
3. Utilizar procesos ágiles y herramientas de software para llevar a cabo el desarrollo del sistema.
4. Diseñar una solución segura que permita proteger la información personal de los clientes.
5. Definir la estructura de costos que implicará la realización de la propuesta planteada.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

### III. MARCO METODOLÓGICO

#### 3.1. TIPO DE INVESTIGACIÓN

Desarrollo de software

#### 3.2. HIPÓTESIS

El desarrollo de una web service ayudará a satisfacer al ciudadano leonardino en cuanto al acceso de información del área de rentas en la Municipalidad Distrital de José Leonardo Ortiz.

#### 3.3. VARIABLES

##### 3.3.1. VARIABLE INDEPENDIENTE

VARIABLE INDEPENDIENTE	DESCRIPCIÓN
Desarrollo de una web service	Aplicativo que permite la comunicación entre programas que han sido creados en distintos lenguajes.

*Tabla 1: Variable independiente  
Fuente: Elaboración propia*

##### 3.3.2. VARIABLE DEPENDIENTE

VARIABLE DEPENDIENTE	DESCRIPCIÓN
Acceso a la información del área de rentas	Sistema encargado de brindar información a toda la ciudadanía del distrito de José Leonardo Ortiz, utilizando la tecnología de Web Service basado en la plataforma PHP.

*Tabla 2: Variable dependiente  
Fuente: Elaboración propia*

## **CAPÍTULO IV**

### **MARCO TEÓRICO**

## IV. MARCO TEÓRICO

### 4.1. ANTECEDENTES DE LA INVESTIGACIÓN

#### 4.1.1. ANTECEDENTES EN EL CONTEXTO INTERNACIONAL

✚ **Título:** “Los sitios web como servicios de información al ciudadano: Un estudio sobre los 308 ayuntamientos de Portugal”.

✚ **Autores:** Eduardo Alfredo Cardoso de Miranda y Antonio Muñoz Cañavate.

✚ **Año de publicación:** 2015

✚ **Lugar:** Universidade Lusófona de Humanidades e Tecnologias. Lisboa – Portugal.

✚ **Conclusión:**

Eduardo Alfredo Cardoso de Miranda y Antonio Muñoz Cañavate concluyeron respecto a los servicios de información, el volcado de información –de todo tipo- en los portales de las administraciones públicas en internet, ha conformado la fase inicial de la administración electrónica en todo el mundo. La facilidad en desarrollar un sistema en el que la información como materia, con una mayor o menor elaboración, estuviera a disposición de los ciudadanos ha facilitado el establecimiento de estos servicios de información a través de los portales de una manera rápida. En este estudio los servicios de información presentan porcentajes muy elevados (por encima del 90%) en los ámbitos de “administración a distancia” y “participación política”, lo que implica una aceptación, por parte del poder local de la necesidad de acercarse a los ciudadanos, y elevados en el ámbito de la vida cotidiana, en torno a un 70% de los indicadores examinados.

✚ **Título:** “Implementar el consumo de servicios vía web services de la información almacenada en una base de datos de una biblioteca digital perteneciente a una institución educativa de nivel medio” – Guayaquil – Ecuador.

✚ **Autor:** Jhon Edison Brito Moreno

✚ **Año de publicación:** 2011

✚ **Lugar:** Universidad de Guayaquil – Ecuador.



### ✚ **Conclusión:**

El autor con esta implementación ayudó a completar un marco tecnológico adecuado, optimizando la educación media, mediante la utilización correcta de las TICs. El objetivo final en este proyecto, es el de automatizar el proceso de acceso a la información, para que los estudiantes puedan realizar sus tareas de una manera efectiva, eficaz y eficiente. Los beneficiarios con este tipo de propuestas, fueron indudablemente los estudiantes y por ende, las instituciones educativas de nivel medio, ya que van a disponer de las herramientas adecuadas para maximizar la utilización de los recursos tecnológicos disponibles en la actualidad. Este proyecto tiene mucho futuro, ya que su área de influencia es un campo donde no existen este tipo de implementaciones en la actualidad, se menciona, además, que este proyecto puede personalizarse agregándole nuevos métodos y utilidades, según las necesidades individuales de cada institución educativa de nivel medio que lo quiera implementar.

#### 4.1.2. ANTECEDENTES EN EL CONTEXTO NACIONAL

✚ **Título:** “Sistema de información web para la mejora de la gestión de la subgerencia de abastecimiento de la Municipalidad Provincial de Utcubamba”.

✚ **Autor:** Juan Carlos Rodríguez Chávez

✚ **Año de publicación:** 2016

✚ **Lugar** Universidad Nacional de Trujillo - Trujillo

### ✚ **Conclusión:**

Según Juan Carlos Rodríguez Chávez se mejoró la gestión de la subgerencia de abastecimiento de la Municipalidad de Utcubamba.

El nivel de satisfacción de los usuarios (entre un rango de 1- 4) con respecto al sistema actual es de 2.00 puntos (50.00%) y con el sistema de información web propuesto es de 3.38 puntos (84.50%), lo que representa un aumento de 1.38 puntos (34.50%), a través de la funcionalidad, control de acceso a la información e interfaz amigable.

✚ **Título:** “Implementación de un web services para mejorar la calidad del servicio aplicado a un sistema ERP del grupo COOPSOL” – Lima.

✚ **Autor:** César Edinson Aguirre Rodríguez

✚ **Año de publicación:** 2012

✚ **Lugar:** Universidad Nacional de Trujillo - Trujillo

✚ **Conclusión:**

César Edinson Aguirre Rodríguez al finalizar su investigación pudo afirmar que la implementación del Web Services le permitió reducir los tiempos en el proceso de emisión de información laboral del área de recursos humanos, así mismo, una mayor fluidez de la información que tendrá una repercusión muy favorable en la organización, porque se eliminarán los costos innecesarios ocasionados por los tiempos muertos que no agregan valor.

#### 4.1.3. ANTECEDENTES EN EL CONTEXTO LOCAL

✚ **Título:** “Portafolio electrónico para facilitar la información de la formación académica de los estudiantes de la Escuela Profesional de Ingeniería en Computación e Informática de la Universidad Nacional Pedro Ruiz Gallo” – Lambayeque 2014.

✚ **Autor:** Luis Jonatan Severino Saenz y Victor Ivan Requejo Delgado

✚ **Año de publicación:** 2014

✚ **Lugar:** Universidad Nacional Pedro Ruiz Gallo – Lambayeque

✚ **Conclusión:**

Los autores concluyeron que con la implementación del portafolio electrónico se genera un mejor desempeño tanto de los docentes como de los estudiantes en cuanto a la compartición de información.

✚ **Título:** “Desarrollo de una aplicación web para la administración de la información del centro oftalmológico Dr. Martin Inoquio Oliva”

✚ **Autor:** Jenny del Pilar Javier Paredes

✚ **Año de publicación:** 2011

✚ **Lugar:** Universidad Nacional Pedro Ruiz Gallo – Lambayeque

✚ **Conclusión:**

Jenny del Pilar concluyó que durante el proceso de implementación de la aplicación web facilitó a ayudar a los clientes y satisfacer sus necesidades para la administración de la información, abarcando las áreas de Consulta, Farmacia y Cirugías del Centro Oftalmológico y la principal característica es que brinda un gran aporte al área de salud, permitiendo registrar

información de la atención medica de los pacientes y obtener estadísticas que permitirán la toma de decisiones del personal médico.

✚ **Título:** “Desarrollo e implementación de un sistema informático la metodología Scrum para optimizar los procesos en el área de ventas y almacén de la botica San Juan – Distrito de Pátapo Chiclayo - Lambayeque”

✚ **Autor:** Ademir César Alfonso Díaz Acuña y Celia Eliana Díaz Vásquez

✚ **Año de publicación:** 2011

✚ **Lugar:** Universidad Nacional Pedro Ruiz Gallo – Lambayeque

✚ **Conclusión:**

Los autores concluyeron que el utilizar la metodología Scrum les permitió administrar los recursos económicos, humanos y llevar un seguimiento del proyecto, también sirvió para calcular la velocidad con que iba el avance de las versiones entregables, flexible a los cambios.

## 4.2. BASE TEÓRICA

### 4.2.1. ANÁLISIS

#### 4.2.1.1. SCRUM

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se

necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente prioriza los objetivos balanceando el valor que le aportan respecto a su coste y quedan repartidos en iteraciones y entregas. De manera regular el cliente puede maximizar la utilidad de lo que se desarrolla y el retorno de inversión mediante la replanificación de objetivos del producto, que realiza durante la iteración con vista a las siguientes iteraciones. (Albaladejo, 2011)

Según (Trigas Gallegos, 2011) los elementos de la metodología Scrum son:

### **Elementos**

- El cliente: Será la persona que nos está pidiendo una solución para un problema específico.
- El usuario: La persona que utilizará esta nueva solución.
- El tiempo: El proyecto se atenderá a unas fechas de comienzo y unas fechas de fin.
- Jefe de Proyecto: Figura necesaria para la gestión de los recursos, así como la planificación del proyecto.

Para poder realizar una buena interrelación de todos estos elementos se debería tener en cuenta una serie de puntos:

- Deberíamos definir qué estructura de organización se va a usar. Es importante en este punto, la aplicación de la experiencia en proyectos anteriores, puesto que, una buena comunicación entre los diferentes departamentos que la definen es vital en la gestión del proyecto.

- Cuando se defina la organización a usar, se tendrá especial cuidado en que pueda cambiar a lo largo del tiempo, es necesario realizar revisiones periódicas.
- Se deberían de definir qué fases van a componer el proyecto.
- Creación de roles y las responsabilidades que implica cada uno de ellos.
- Las tareas que definen el proyecto deberían de estar definidas de forma correcta. Al mismo tiempo, las tareas tienen que ser validadas por el personal responsable.
- Gestionar todos estos elementos necesitan una figura – sería aquí donde aparece el Jefe de Proyectos – y también una entidad dentro de la organización que sería la Oficina de Dirección de Proyectos.

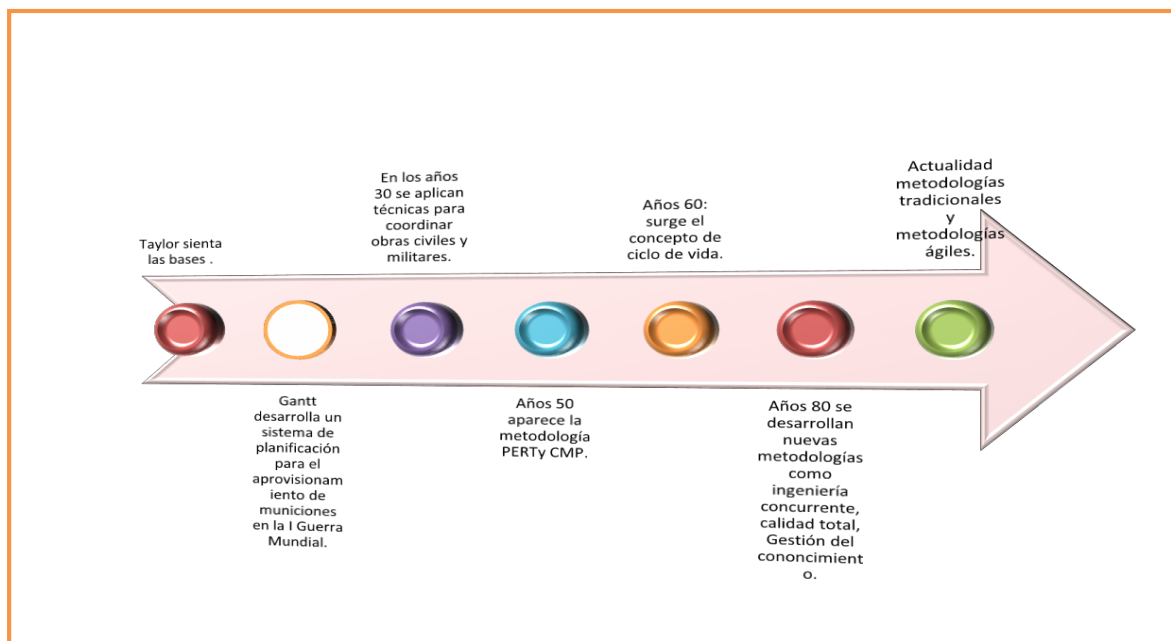


Figura 2: Evolución histórica de la gestión de proyectos  
Autor: Manuel Trigás Gallegos

Todos estos puntos llevan a la aparición de una nueva disciplina, la Gestión de Proyectos.

Realmente la Gestión de Proyectos surgió ya como necesidad en los años 50 en el ámbito militar. Los proyectos que se desarrollaban eran de gran magnitud y necesitaban de personal cualificado en diferentes disciplinas, por lo tanto, la coordinación de estos grupos era un paso natural.

A partir de este momento surgen nuevos conceptos a la par que herramientas que van a facilitar la gestión de un proyecto, como son el ciclo de vida, descomposición en tareas, realización de gráficos, etc.

La Gestión de Proyectos se podría definir como “la disciplina de planear, organizar, asegurar y coordinar recursos y personas para cumplir con los Objetivos, Entregables y Criterios de Éxito de los proyectos”.

Una breve descripción de la evolución histórica de la gestión de proyectos podríamos mostrarla en la siguiente gráfica:

### **Fases y ciclos de vida de un proyecto**

El comienzo de un proyecto siempre viene acompañado de un grado de incertidumbre. Como opción para paliarlo se optará por anidar las tareas a realizar en diferentes fases. (Trigas Gallegos, 2011)

Estas fases servirán para tener un control más específico de cada una de las tareas, y formarán el denominado Ciclo de Vida de un Proyecto.

Los ciclos de vida de un proyecto servirán para definir el comienzo y el final del mismo. Generalmente cuando una empresa emprende un proyecto lo primero que solicitará será un estudio de viabilidad o factibilidad. Este procedimiento puede ser la base para que se marque un inicio del proyecto.

De la misma manera, el ciclo de vida se usa como vínculo con las distintas tareas que se realizan dentro de la empresa u organización.

Las fases definidas por la mayoría de los ciclos de vida de proyectos en general incluyen alguna forma de transferencia de tecnología, tales como requisitos a diseño, construcción a operaciones o de diseño a implementación. Al finalizar una fase se entregará una documentación, proceso, etc... que comúnmente se denomina entregable, que será revisada y aceptada antes de continuar con la siguiente fase; además servirá para corregir y detectar los posibles errores que se vayan produciendo, así como las desviaciones en los costes. Las evaluaciones de cada final de fase suelen tener diferentes nombres como salidas de fase, puntos de impacto o puertas de etapas.

Sin embargo, en algunas ocasiones se comienza la fase siguiente antes de la aprobación de los productos a entregar de la fase anterior, cuando los riesgos percibidos se pueden asumir. A la práctica de superponer fases suele denominarse vía rápida.

La mayor parte de los ciclos de vida comparten las siguientes características:

- El coste del personal es bajo al comienzo, más alto hacia la mitad, y menor al llegar al final del proyecto.
- El riesgo y la incertidumbre de finalizar el proyecto con éxito es más alta al principio del proyecto y la probabilidad de fracaso disminuye a medida que se va avanzando en el proyecto.
- La influencia por parte de los participantes en el producto es más alta al comienzo del mismo y decae progresivamente con la continuación del proyecto. Hay que tener en cuenta que el coste de los cambios y los errores tiende a aumentar a medida que avanza el proyecto.

Los ciclos de vida del proyecto responden a:

- ¿Qué trabajo se va a realizar en cada fase?
- ¿Quién participará en cada fase?

Las fases se definirán de forma secuencial, es decir, que una fase no comienza hasta que termina la otra. Suelen contener una serie hitos o tareas que marcan los momentos más importantes en el desarrollo del proyecto.

Así pues, la base de cada fase se puede resumir en:

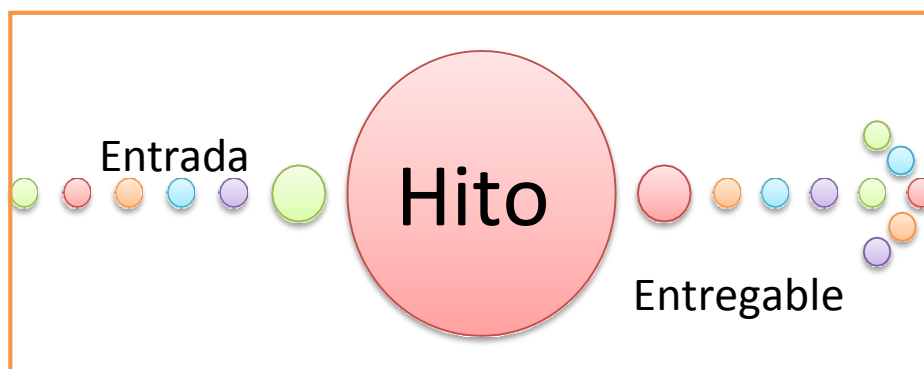


Figura 3: Elementos básicos que formarán una fase  
Autor: Manuel Trigás Gallegos

### ¿Por qué utilizar Scrum?

Utilizar Scrum nos ayudará a realizar entregas a corto plazo según los requisitos más prioritarios en ese momento para una mejor flexibilidad y adaptación respecto a las necesidades del cliente o cambios en el mercado y así disminuir los riesgos.

	<b>SCRUM</b>	<b>RUP</b>
<b>Enfoque</b>	Iterativo	Iterativo
<b>Ciclo de desarrollo</b>	Cada sprint (iteración) es un ciclo completo	Ciclo formal se define a través de 4 fases, pero algunos flujos de trabajo pueden ser concurrentes
<b>Planificación</b>	Cada plan de la siguiente iteración se determina al final de la iteración actual	Plan de proyecto formal, asociada a múltiples iteraciones
<b>Alcance</b>	Es definido en la lista de objetos (Project backlog) que es reevaluado al fin de cada iteración (sprint)	Es definido antes del comienzo del proyecto y plasmado en el documento de alcance
<b>Artefactos (documentos / entregables)</b>	El software operativo es el único artefacto	Documento de alcance, visión, caso de negocio, lista de riesgos, plan de desarrollo, plan de iteraciones, lista de principales casos de uso, etc.
<b>Tipo de proyecto</b>	Proyectos con requisitos cambiantes o implementación de mejoras rápidas. Equipos auto organizados y cross funcionales	Proyectos empresariales de largo alcance y tiempo

Tabla 3: SCRUM vs. RUP  
Fuente: Elaboración propia



## **4.2.2. HERRAMIENTAS DE DESARROLLO**

### **4.2.2.1. XAMPP**

Según (Carrión, Noriega, & Del Castillo, 2019), sostiene que XAMPP es un servidor independiente de plataforma de código libre. El cual nos permite instalar de manera sencilla Apache en nuestro propio ordenador, sin importar el sistema operativo (Linux, Windows, MAC o Solaris). Y lo mejor de todo es que su uso es gratuito.

XAMPP incluye además servidores de bases de datos como MySQL y SQLite con sus respectivos gestores phpMyAdmin y phpSQLiteAdmin. Incorpora también el intérprete de PHP, el intérprete de Perl, servidores de FTP como ProFTPD ó FileZilla FTP Serve, etc. entre muchas cosas más.

Es una herramienta de desarrollo que te permite probar tu trabajo (páginas web o programación, por ejemplo) en tu propio ordenador sin necesidad de tener que acceder a internet.

#### **Paquetes básicos:**

- Apache, el servidor Web más famoso.
- MySQL, una excelente base de datos de código libre.
- PHP y Perl: lenguajes de programación.
- ProFTPD: un servidor FTP.
- OpenSSL: para soporte a la capa de sockets segura.

#### **Paquetes gráficos:**

- GD (Graphics Draw): la librería de dibujo de gráficos.
- libpng: la librería oficial de referencia de PNG.
- libjpeg: la librería oficial de referencia de JPEG.
- ncurses: la librería de gráficos de caracteres.

#### **Paquete de bases de datos:**

- gdbm: la implementación GNU de la librería standard dbm de UNIX.
- SQLite: un motor de base de datos SQL muy pequeño y cero configuraciones.

- FreeTDS: una librería de base de datos que da a los programas de Linux y UNIX la habilidad de comunicarse con Microsoft SQL y Sybase.

#### **Paquetes XML:**

- expat: una librería parser de XML.
- Salbotron: una toolkit de XML.
- libxml: un parser C de XML y un toolkit para GNOME.

#### **Paquetes PHP:**

- PEAR: la librería de PHP.
- Una clase pdf que genera documentos PDF dinámicos con PHP.
- TURCK MMCache: un potenciador de la performance de PHP.

#### **Otros paquetes:**

- zlib: una librería de compresión.
- mod\_perl: empotra un intérprete de Perl en Apache.
- gettext: un conjunto de herramientas que asiste a los paquetes GNU para producir mensajes multilinguales.
- mcrypt: un programa de encriptación.
- Ming: una librería de salida en Flash.
- IMAC C-Client: un API de correos

#### **4.2.2.2. SQL SERVER 2012**

Es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. (Rouse, 2015)

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.

- Además, permite administrar información de otros servidores de datos.

#### 4.2.2.3. PHP (PHP Versión 7.2.6)

Como abreviatura recurrente en inglés de PHP: Hypertext Precrocesor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el HTML resultante.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo. (Grupo PHP, 2019)

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3\_01, es una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, esta licencia no tiene restricciones de copyleft asociadas con GPL.

- ✓ Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- ✓ Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- ✓ El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.

- ✓ Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- ✓ Capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones).
- ✓ Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Permite aplicar técnicas de programación orientada a objetos.
- ✓ No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- ✓ Tiene manejo de excepciones (desde PHP5).
- ✓ Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador<sup>37</sup> puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- ✓ Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

#### **4.2.2.4. POSTMAN**

Según (Palacios Tolón & Moreno Sama, 2019) Postman es una herramienta dirigida a desarrolladores web que permite realizar peticiones HTTP a cualquier API. Postman es muy útil a la hora de programar y hacer pruebas, puesto que nos ofrece la posibilidad de comprobar el correcto funcionamiento de nuestros desarrollos.

Con esto no queremos decir que Postman sea una herramienta exclusiva para profesionales del entorno web, de hecho, va a ser muy útil para todo aquel que tenga que interactuar con una API.

Imagínate que un día te levantas curioso y quieres ver qué hace la API de Spotify. Si navegas a su web podrás encontrar documentación, pero comprenderás más y mejor cómo funciona y qué te ofrece si lo pruebas por ti mismo.

Fuera de su objetivo principal, hacer peticiones a servicios, nos ofrece un conjunto de funcionalidades que nos ayudarán a organizar las peticiones en colecciones, hacer y automatizar pruebas, mantener equipos sincronizados y crear Mocks de APIs.

Todo esto es gratis en su versión básica pero además ofrece dos posibilidades de pago que mejoran las características, permitiendo la sincronización entre miembros de un equipo, creación de Mocks sin límites, monitorización de entornos, etc.

- ✓ Automatización de pruebas te permiten ejecutar conjuntos de pruebas, más o menos complejas.
- ✓ Además, si defines Test específicos en las request, podrás hacer pruebas más complejas que reproduzcan la combinación de varias operaciones. Al ejecutarlo, mostrará el resultado de las pruebas con un resumen, indicando el tiempo de respuesta y el estado HTTP devuelto por el servidor.
- ✓ Variables de entorno, las variables son símbolos que guardan valores y para esto mismo sirven en Postman. Aunque a priori pueda parecerte que no tiene mucho sentido para una herramienta así, lo cierto es que esto aumenta su potencial enormemente. El uso de variables nos ayuda a la realización de pruebas, al cambio entre distintos entornos (entorno de desarrollo, pruebas, producción, etc.) y, en general, nos ayuda a recordar y evitar escribir el mismo texto o ruta de forma repetitiva.

#### **4.2.2.5. VISUAL STUDIO CODE**

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código. También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo una licencia propietaria.

Visual Studio Code se basa en Electron, un marco que se usa para implementar aplicaciones Node.js para el escritorio que se ejecuta en el motor de diseño Blink. Aunque utiliza el marco de trabajo de Electron, el software no usa Atom y en su lugar emplea el mismo componente de editor (nombre en código "Mónaco") que se usa en Visual Studio Team Services (anteriormente llamado Visual Studio Online).

En la Encuesta de Desarrolladores de Stack Overflow 2018, Visual Studio Code fue clasificada como la herramienta de entorno de desarrollador más popular, con un 34.9% de 75,398 encuestados que afirmaron usarla.

- ✓ Visual Studio Code es un editor de código fuente. Admite varios lenguajes de programación y un conjunto de funciones que pueden estar disponibles o no para un lenguaje determinado.
- ✓ Muchas de las funciones de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a ellos a través de la paleta de comandos o a través de un archivo json (por ejemplo, las preferencias del usuario).
- ✓ La paleta de comandos es una interfaz de línea de comandos. Sin embargo, desaparece si el usuario hace clic en cualquier lugar fuera de él o presiona una combinación de teclas en el teclado para interactuar con algo fuera de él. Cuando esto sucede, el comando en curso se cancela.
- ✓ En la función de un editor de código fuente, Visual Studio Code permite cambiar la página de códigos en la que se guarda el documento activo, el carácter que identifica el salto de línea (una

opción entre LF y CRLF) y el lenguaje de programación del documento activo.

#### **4.2.2.6. FRAMEWORK CODEIGNITER 3.9**

(Fontán, 2012) CodeIgniter es un framework para el desarrollo de aplicaciones en PHP que utiliza el MVC. Permite a los programadores Web mejorar la forma de trabajar y hacerlo a mayor velocidad.

Al igual que cualquier framework está pensado para gente que tiene un dominio, al menos medio, del lenguaje de programación PHP. Siempre hay que controlar PHP “a pelo” para empezar a trabajar de forma eficiente con este framework (o cualquier otro).

- ✓ Las páginas se procesan más rápido, el núcleo de CodeIgniter es bastante ligero.
- ✓ Es sencillo de instalar, basta con subir los archivos al FTP y tocar un archivo de configuración para definir el acceso a la base de datos.
- ✓ Reutilización de código, desarrollo ágil.
- ✓ Existe abundante documentación en la red.
- ✓ Facilidad de edición del código ya creado.
- ✓ Facilidad para crear nuevos módulos, páginas o funcionalidades.
- ✓ Acceso a librerías públicas y clases. Entre otras, hay librerías para el login, paginador, calendarios, fechas.
- ✓ Estandarización del código. Fundamental cuando hay que tocar código hecho por otra persona o cuando trabaja más de una persona en un mismo proyecto.
- ✓ URLs amigables con SEO. Hoy en día creo que nadie duda de la importancia del posicionamiento web.
- ✓ Separación de la lógica y arquitectura de la web, el MVC.
- ✓ CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar, pero podemos seguirla o no (esto puede convertirse en un inconveniente también)
- ✓ Cualquier servidor que soporte PHP+MySQL sirve para CodeIgniter.
- ✓ CodeIgniter se encuentra bajo una licencia open source, es código libre.

- ✓ CodeIgniter usa una versión modificada del Patrón de Base de Datos Active Record. Este patrón permite obtener, insertar y actualizar información in tu base de datos con mínima codificación. Permite queries más seguras, ya que los valores son escapadas automáticamente por el sistema.

#### **4.2.2.7. ANDROID STUDIO (API 27)**

(Academia Android, 2015) Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, MacOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Estuvo en etapa de vista previa de acceso temprano a partir de la versión 0.1, en mayo de 2013, y luego entró en etapa beta a partir de la versión 0.8, lanzada en junio de 2014. La primera compilación estable, la versión 1.0, fue lanzada en diciembre de 2014. La última versión estable es la 3.0, y fue lanzada en octubre de 2017.

- ✓ Integración de ProGuard y funciones de firma de aplicaciones.
- ✓ Renderizado en tiempo real
- ✓ Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- ✓ Soporte para construcción basada en Gradle.
- ✓ Refactorización específica de Android y arreglos rápidos.
- ✓ Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario.
- ✓ Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- ✓ Plantillas para crear diseños comunes de Android y otros componentes.



- ✓ Soporte para programar aplicaciones para Android Wear.
- ✓ Soporte integrado para Google Cloud Platform, que permite la integración con Google Cloud Messaging y App Engine.
- ✓ Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.

### Requerimientos del sistema

Windows	Mac OS	Linux
Microsoft Windows 8/7/Vista/2003 (32 o 64 bit)	Mac OS X 10.8.5 o superior, hasta la 10.9 (Mavericks)	GNOME o entorno de escritorio KDE
Mínimo de 2 GB de RAM, recomendado 4 GB de RAM		
400 MB de espacio en disco		
Necesita de al menos 1 GB para Android SDK, emulador de imágenes del sistema, y cachés		
Resolución mínima de pantalla de 1280 x 800		
Java Development Kit (JDK) 7 o superior		

*Figura 4: Requerimientos del Sistema  
Autor: Academia Android*

Para los sistemas Linux será necesaria la biblioteca de C GNU (glibc) 2.11 o posterior. Además, para los sistemas Mac OS será necesario ejecutar Android Studio con Java Runtime Environment (JRE) 6 para la renderización optimizada de fuentes.

### Comparativa entre Android Studio y ADT Eclipse

Para una mayor comprensión de las diferencias y novedades que presenta Android Studio con respecto al IDE Eclipse, y más concretamente con el ADT para Android, se propone la siguiente tabla comparativa entre ambas opciones:

Características	Android Studio	ADT
Sistema de construcción	Gradle	ANT
Construcción y gestión de proyectos basados en Maven (herramienta de software para la gestión y construcción de proyectos Java, similar a Apache ANT, pero su modelo es más simple ya que está basado en XML)	Si	No (es necesario instalar un plugin auxiliar)
Construir variantes y generación de múltiples APK (muy útil para Android Wear)	Si	No
Refactorización y completado avanzado de código Android	Si	No
Diseño del editor gráfico	Si	Si
Firma APK y gestión de almacén de claves	Si	Si
Soporte para NDK (Native Development Kit: herramientas para implementar código nativo escrito en C y C++)	Próximas versiones	Si
Soporte para Google Cloud Platform	Si	No
Vista en tiempo real de renderizado de layouts	Si	No
Nuevos módulos en proyecto	Si	No
Editor de navegación	Si	No
Generador de assets	Si	No
Datos de ejemplo en diseño de layout (sin renderizar en tiempo de ejecución)	Si	No
Visualización de recursos desde editor de código	Si (a la izquierda de la línea de asignación del recurso)	No

Figura 5: Comparativa entre Android Studio y ADT Eclipse  
Autor: Academia Android

#### 4.2.2.8. MICROSOFT AZURE

Azure es una nube pública de pago por uso que te permite compilar, implementar y administrar rápidamente aplicaciones en una red global de Data Centers (centros de datos) de Microsoft.

El significado de nube pública, no quiere decir que cualquiera pueda acceder a la información de tu negocio, sino que cualquiera con conexión a Internet puede hacer uso de esta nube y que sólo en caso de disponer de las credenciales oportunas, podrá acceder a tu portal de Azure, donde se generan y administran todos los servicios.

En el portal Microsoft Azure existen diferentes servicios de infraestructura y de plataforma para que puedas montar los servicios que necesites de manera sencilla, con unos cuantos clics. En tu portal dispones de un botón [New +] y a partir de ahí, sólo tienes que elegir la región de los Data Center donde estará tu servicio y, a continuación, el tipo de servicio con sus características. Por ejemplo, en el caso de querer crear una máquina virtual podrías seleccionar el tipo de máquina como Windows Server 2016 Data Center, a continuación, rellenar todas sus características (espacio de disco, RAM, etc.). En este proceso de creación, se define un nombre de usuario y contraseña específica para poder iniciar sesión en esta máquina virtual.

Entre los servicios, dispones de infraestructuras (IaaS: almacenamiento, redes, máquinas virtuales, etc.) y plataformas (PaaS: bases de datos de alta disponibilidad SQL, CMS para desarrollo de web, backend para aplicaciones móviles, etc.). Son compatibles con todo tipo de tecnología: bases de datos Oracle, Linux, php, iOS, My SQL, Android, PHP.

Estos servicios están garantizados con una disponibilidad del 99.99 %, y en caso de fallo en disponibilidad superior, Microsoft se compromete a indemnizar por los daños. Además, cuenta con todas las certificaciones en materia seguridad y protección de datos. ¿Sabías que es la primera plataforma cloud que cumple en categoría alta las características de la certificación de conformidad del Esquema Nacional de Seguridad?

**Grandes ventajas: Flexibilidad y pago por uso. ¿Cuándo es realmente útil?**

Esta potente solución empresarial para la gestión de tus infraestructuras y aplicaciones presenta grandes ventajas: destacamos la flexibilidad y el pago

por uso. Con Azure ya no es necesaria la inversión en máquinas físicas y el mantenimiento que conllevan. De esta manera, se simplifica el despliegue ya que no hay que comprometerse con largos periodos de amortización, se agiliza el proceso de configuración y nos permite adaptar nuestros sistemas a nuestras necesidades en pocos minutos, con total flexibilidad. Además, dispone de herramientas para poder monitorizar el rendimiento y así predecir si es necesario escalar o no nuestras máquinas, y, en consecuencia, pagar por lo que realmente se necesita. Por este motivo, distinguimos tres situaciones en las que, sin duda, recomendamos utilizar los servicios de Azure:

- ✓ En el despliegue de una solución en la que el incremento de uso va a ser exponencial (o desconocido) y no podemos definir con garantías el pico de demanda máximo.
- ✓ Cuando sabes que la demanda del servicio va a fluctuar en el tiempo.
- ✓ En entornos de desarrollo o pruebas que posteriormente podrían seguirse utilizando o no, o que pueden “apagarse” puntualmente y después volverse a activar.

Y, por último, también lo recomendamos como solución de backup o site recovery (continuidad de negocio) por la seguridad extra que aporta a la información de tu negocio a un precio muy competitivo. Por ejemplo, una solución de backup de 50 GB sólo costaría 5,23€/mes, y de 500 GB 18,53€. (Tecon, 2019).

### **4.2.3. TECNOLOGÍA DE DESARROLLO DE LA WEB SERVICE**

#### **4.2.3.1. JWT TOKEN**

Según (Paszniuk, 2016) Json Web Token es un conjunto de medios de seguridad para peticiones HTTP y así representar demandas para ser transferidos entre dos partes (cliente y servidor). Las partes de un JWT se codifican como un objeto JSON que está firmado digitalmente utilizando JSON Web Signature (JWS).

La mayoría de las aplicaciones actuales consumen servicios rest y están alojadas en distintos dominios con lo cual no podemos trabajar con sesiones ya que se almacenan en este.

Podemos decir que la mejor alternativa es llevar a cabo la autenticación haciendo uso de tokens que vayan del servidor al cliente, un usuario hace login (no necesita enviar token porque no lo tiene), una vez el servidor de ok retorna un token cómo respuesta y el usuario debe enviar dicho token en las siguientes peticiones para poder acceder a los recursos del servicio.

En cada petición el servidor debe comprobar el token proporcionado por el usuario y si es correcto podrá acceder a los recursos solicitados, de otra forma deberá denegar la petición.

También nos añade más seguridad. Al no utilizar cookies para almacenar la información del usuario, podemos evitar ataques CSRF (Cross-Site Request Forgery) que manipulen la sesión que se envía al backend. Por supuesto podemos hacer que el token expire después de un tiempo lo que le añade una capa extra de seguridad.

El formato de un JWT está compuesto por 3 strings separados por un punto . algo así como:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI1NGE4Y2U2MThlOTFiMGlxMzY2NWUyZjkiLCJpYXQiOiIxNDI0MTgwNDg0IiwiaXhwIjoIMTQyNTM5MDE0MiJ9.yk4nouUteW54F1HbWtgg1wJxeDjqDA_8AhUPyjE5K0U
```

*Tabla 4: Formato de un JWT  
Autor: Rodrigo Paszniuk*

Cada string significa una cosa:

- **Header:** La primera parte es la cabecera del token, que a su vez tiene otras dos partes, el tipo, en este caso un JWT y la codificación utilizada. Comúnmente es el algoritmo HMAC SHA256, el contenido sin codificar es el siguiente:

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

*Tabla 5: Contenido Header  
Autor: Rodrigo Paszniuk*

**Su codificación sería:** eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9

- **Payload:** El Payload está compuesto por los llamados JWT Claims donde irán colocados los atributos que definen nuestro token. Existen varios que se pueden consultar aquí, los más comunes a utilizar son:
  - **sub:** Identifica el sujeto del token, por ejemplo: un identificador de usuario.
  - **iat:** Identifica la fecha de creación del token, válido para si queremos ponerle una fecha de caducidad. En formato de tiempo UNIX.
  - **exp:** Identifica a la fecha de expiración del token. Podemos calcularla a partir del iat. También en formato de tiempo UNIX.

```
{
  "sub": "54a8ce618e91b0b13665e2f9",
  "iat": "1424180484",
  "exp": "1425390142"
}
```

*Tabla 6: Contenido Payload I  
Autor: Rodrigo Paszniuk*

#### Su codificación sería:

eyJzdWl0eS4a8ce618e91b0b13665e2f9eyJpYXQiOiIxNDI0MTgwNDg0IiwiaWF0IjoiMTQyNTM5MDE0IiwiaGVhZGVzIjoiIn0=

También podemos añadirle más campos, incluso personalizados, como pueden ser el rol del usuario, etc.

```
{
  "sub": "54a8ce618e91b0b13665e2f9",
  "iat": "1424180484",
  "exp": "1425390142",
  "role": 2
}
```

*Tabla 7: Contenido Payload II  
Autor: Rodrigo Paszniuk*

- **Signature:** La firma es la tercera y última parte del JSON Web Token. Está formada por los anteriores componentes (Header y Payload)

cifrados en Base64 con una clave secreta (almacenada en nuestro backend). Así sirve de Hash para comprobar que todo está bien.

```

HMACSHA256(
    base64UrlEncode(header) + "." +
    base64UrlEncode(payload), secret
);

```

Tabla 8: Contenido Signature  
Autor: Rodrigo Paszniuk

**Su codificación sería:**

yk4 yk4nouUteW54F1HbWtgg1wJxeDjqDA\_8AhUPyjE5K0U

Por tanto, todo nuestro JSON Web Token, una vez codificado tendrá esta codificación:

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI1NGE4Y2U2MTI0OTFmMGlxMzY2NWUyZjkiLCJpYXQiOiIxNDI0MTgwNDg0IiwiaXhwaIjoimjNTM5MDE0MiJ9.yk4nouUteW54F1Hbwtgg1wJxeDjqDA\_8AhUPyjE5K0U

Tabla 9: Contenido JSON Web Token  
Autor: Rodrigo Paszniuk

#### 4.2.3.2. VOLLEY ANDROID

Según (James, 2018) Volley es una librería desarrollada por Google para optimizar el envío de peticiones Http desde las aplicaciones Android hacia servidores externos.

Volley, es un cliente HTTP creado para facilitar la comunicación de red en las aplicaciones Android. A diferencia de la interfaz `HttpURLConnection`, Volley está totalmente enfocado en las peticiones, evitando la creación de código repetitivo para manejar tareas asíncronas por cada petición o incluso para analizar los datos que vienen del flujo externo.

- ✓ Procesamiento concurrente de peticiones.
- ✓ Priorización de las peticiones, lo que permite definir la preponderancia de cada petición.
- ✓ Cancelación de peticiones, evitando la presentación de resultados no deseados en el hilo principal.

- ✓ Gestión automática de trabajos en segundo plano, dejando de lado la implementación manual de un framework de hilos.
- ✓ Implementación de caché en disco y memoria.
- ✓ Capacidad de personalización de las peticiones.
- ✓ Provee información detallada del estado y flujo de trabajo de las peticiones en la consola de depuración.

#### 4.2.3.3. JSON

Es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

En la práctica, los argumentos a favor de la facilidad de desarrollo de analizadores o de sus rendimientos son poco relevantes, debido a las cuestiones de seguridad que plantea el uso de `eval()` y el auge del procesamiento nativo de XML incorporado en los navegadores modernos. Por esa razón, JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia (de aquí su uso por Yahoo, Google, etc., que atienden a millones de usuarios) cuando la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente.

Si bien es frecuente ver JSON posicionado contra XML, también es frecuente el uso de JSON y XML en la misma aplicación. Por ejemplo, una aplicación de cliente que integra datos de Google Maps con datos meteorológicos en SOAP hacen necesario soportar ambos formatos.

En diciembre de 2005 Yahoo! comenzó a dar soporte opcional de JSON en algunos de sus servicios web.



```

{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitem": [
        {
          "value": "New", "onclick": "CreateNewDoc()"
        }, {
          "value": "Open", "onclick": "OpenDoc()"
        }, {
          "value": "Close", "onclick": "CloseDoc()"
        }
      ]
    }
  }
}

```

Es una posible representación JSON del siguiente XML:

```

<menu id="file" value="File">
  <popup>
    <menuitem value="New" onclick="CreateNewDoc()" />
    <menuitem value="Open" onclick="OpenDoc()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>

```

### Sintaxis de JSON

Para escribir JSON debemos tener en cuenta lo siguiente: La creación de los objetos JSON implica escribir datos, para ello:

- ✓ Los datos están separados por comas.
- ✓ Los datos se escriban en pares, siendo primero el nombre o atributo del mismo y luego el valor del dato.
- ✓ Los objetos JSON están rodeados por llaves “{ }”

- ✓ Llaves cuadradas “[ ]” guardan arreglos, incluyendo otros objetos JSON.

### ¿Por qué utilizar JSON?

Porque permite el intercambio de información entre dos sistemas: Android y SQL server que no comparten el mismo lenguaje.

#### 4.2.3.4. REST\_CONTROLLER

Es una clase creada bajo el patrón MVC para el rápido y fácil acceso de aplicaciones web simples o complejas.

Dándole al desarrollador de API y la aplicación cliente la elección del formato de datos a usar, la API es abierta a una audiencia mucho más amplia y puede ser usado con más lenguajes de programación y sistemas. Estos tres no son los únicos formatos soportados, desde el inicio tu API REST puede usar:

- ✓ **XML:** Casi cualquier lenguaje de programación puede leer xml
- ✓ **JSON:** Útil para JavaScript e incrementalmente en aplicaciones PHP.
- ✓ **CSV:** Se abre con programas de hojas de cálculo
- ✓ **HTML:** una tabla simple de HTML
- ✓ **PHP:** Representación de código PHP que puede ser eval()'uado
- ✓ **SERIALIZE:** Datos serializados que pueden ser deserializados en PHP

Mientras que agregar el formato a la URL no es técnicamente la forma más RESTful de cambiar formatos, permite hacer pruebas fácilmente en el navegador y deja a los desarrolladores sin cURL ejecutar simples solicitudes GET en la API. La forma más RESTful es enviar una cabecera Content-type al controlador REST usando cURL, pero eso será explicado después.

- ✓ **GET:** Obtener información.
- ✓ **POST:** Enviar información y guardar.
- ✓ **PUT:** Enviar información y actualizar.
- ✓ **DELETE:** Eliminar.

**CAPÍTULO V**  
**DESARROLLO DE LA**  
**PROPUESTA**

## **V. DESARROLLO DE LA PROPUESTA**

### **5.1. PLANTEAMIENTO DEL PROYECTO**

#### **5.1.1. IDENTIFICACIÓN DEL ESCENARIO DEL PROYECTO**

Este proyecto tiene como fin ofrecer una herramienta que facilite el acceso de información a los contribuyentes, otorgándoles datos privados de los tributos municipales que se otorgan del área de Renta de la Municipalidad del Distrito de José Leonardo Ortiz – Chiclayo.

Es decir, el proyecto se desarrolla teniendo como escenario la Municipalidad y los requerimientos de esta.

Una de las ventajas de este proyecto es que permite a estas personas controlar, conocer y verificar su estado como contribuyente, así como también una mejor calidad de servicio a sus clientes en aspectos relacionados a costos y tiempos de atención.



*Figura 6: Municipalidad Distrital de José Leonardo Ortiz*

### **5.1.2. IDENTIFICACIÓN DE LOS USUARIOS Y ÁREAS INTERVENIENTES**

Los beneficiarios del presente proyecto son las personas contribuyentes del Distrito de José Leonardo Ortiz. El cliente va a poder acceder directamente a información relacionada con sus deberes (pagos, impuestos, etc.).

Los beneficiarios indirectos son los trabajadores de la Municipalidad, encargados de la entrega y manipulación de dichos datos. Otorga beneficios para ambas personas.

#### **5.1.2.1. USUARIOS**

Las acciones de control de la Web Services permitirá al usuario optimizar el tiempo en las operaciones a realizar, logrando así satisfacer a los ciudadanos del distrito de José Leonardo Ortiz.

Debemos señalar que el hecho del desarrollo de la Web Services, dirigido a los contribuyentes de la Municipalidad no es impedimento para que ellos puedan dirigirse a la Municipalidad específicamente al área de rentas para que se les brinde alguna u otra información que deseen conocer o que no les haya quedado claro en la aplicación del sistema. Así como también los trabajadores de dicha área en casos como el que se menciona deberán seguir con las funcionalidades habituales y/u otras designadas.

### **5.1.3. ANÁLISIS DE FACTIBILIDAD**

#### **5.1.3.1. FACTIBILIDAD OPERATIVA**

El desarrollo de este proyecto de investigación, favorece a los ciudadanos del Distrito de José Leonardo Ortiz de manera directa, con la información relacionada con sus tributos, puesto que la aplicación permite contar con información actualizada, confiable y organizada de tal forma que el acceso se realice en forma eficiente y fácil manejo.

#### **5.1.3.2. FACTIBILIDAD TÉCNICA**

Técnicamente es posible el desarrollo del Sistema, con una organizada programación en PHP y una Web Services para la comunicación entre la base de datos y los sistemas web y móvil (Android).

Existe la suficiente fundamentación para desarrollar la Web Services, obtenidas de sitios web, bibliografías confiables, lo cual ayudará a cubrir con todas las expectativas planteadas a inicio de la elaboración del proyecto.

## **5.2. DEFINICIÓN DE REQUERIMIENTOS DEL NEGOCIO**

### **5.2.1. ANÁLISIS DEL REQUERIMIENTO**

El análisis de requerimientos, es la parte principal del desarrollo del proyecto; aquí es donde se concretó la idea, la visión del sistema y los requerimientos.

La idea se basa en desarrollar un software, para una mejor calidad de servicio a sus clientes en aspectos relacionados a costos y tiempos de atención. Por lo que se propuso el desarrollo de una Web Service para Android en PHP, en la que el cliente pueda acceder directamente a la información relacionada con sus deberes (pagos, impuesto, etc.), ya que con más frecuencia se necesita la comunicación y transferencia de datos entre nuestro dispositivo móvil y los grandes administradores de base de datos.

### **5.2.2. REQUERIMIENTOS FUNCIONALES**

#### **5.2.2.1. IMPLEMENTACIÓN DE LA WEB SERVICES**

Proporciona comunicación entre la base de datos SQL Server y las aplicaciones.

#### **5.2.2.2. DESARROLLO DEL SISTEMA WEB (Tesis\_Muni)**

Permite enviar información de las deudas de los contribuyentes al Sistema de banco, verifica los pagos realizados en él.

#### **5.2.2.3. DESARROLLO DEL SISTEMA WEB (Tesis\_Banco)**

Permite consultar las deudas de los contribuyentes y realizar diferentes pagos del área de rentas de la Municipalidad. Posteriormente al finalizar el día, envía el listado de pagos realizados por los contribuyentes.

#### **5.2.2.4. CONSULTA DE INFORMACIÓN**

Mediante la aplicación móvil se verifica pagos y deudas.

### **5.2.3. REQUERIMIENTOS NO FUNCIONALES**

#### **5.2.3.1. APARIENCIA O INTERFAZ**

El software ha de ser intuitivo y amigable.

#### **5.2.3.2. USABILIDAD**

El software ha de ser fácil de usar para el usuario teniendo en cuenta que estará dirigido a todo tipo de clientes.

#### **5.2.3.3. RENDIMIENTO**

El software debe responder rápidamente a las peticiones del usuario.

#### **5.2.3.4. SOPORTE Y PORTABILIDAD**

El software debe ser instalado en laptops y móviles.

### **5.3. DISEÑO Y DESARROLLO DE PRESENTACIÓN DE DATOS**

A continuación, se muestra cuáles son las tecnologías utilizadas en nuestro proyecto, donde nuestra aplicación Android se pondrá en contacto con nuestro servicio web, el cual, estará desplegado sobre nuestro servidor Apache Xampp. Será este servicio web que realice las consultas a nuestra base de datos SQL Server. Además, en esta fase es donde se planifica las actividades necesarias, especificando las características y el diseño de la arquitectura del Sistema Web y Móvil.

### 5.3.1. DISEÑO ARQUITECTÓNICO

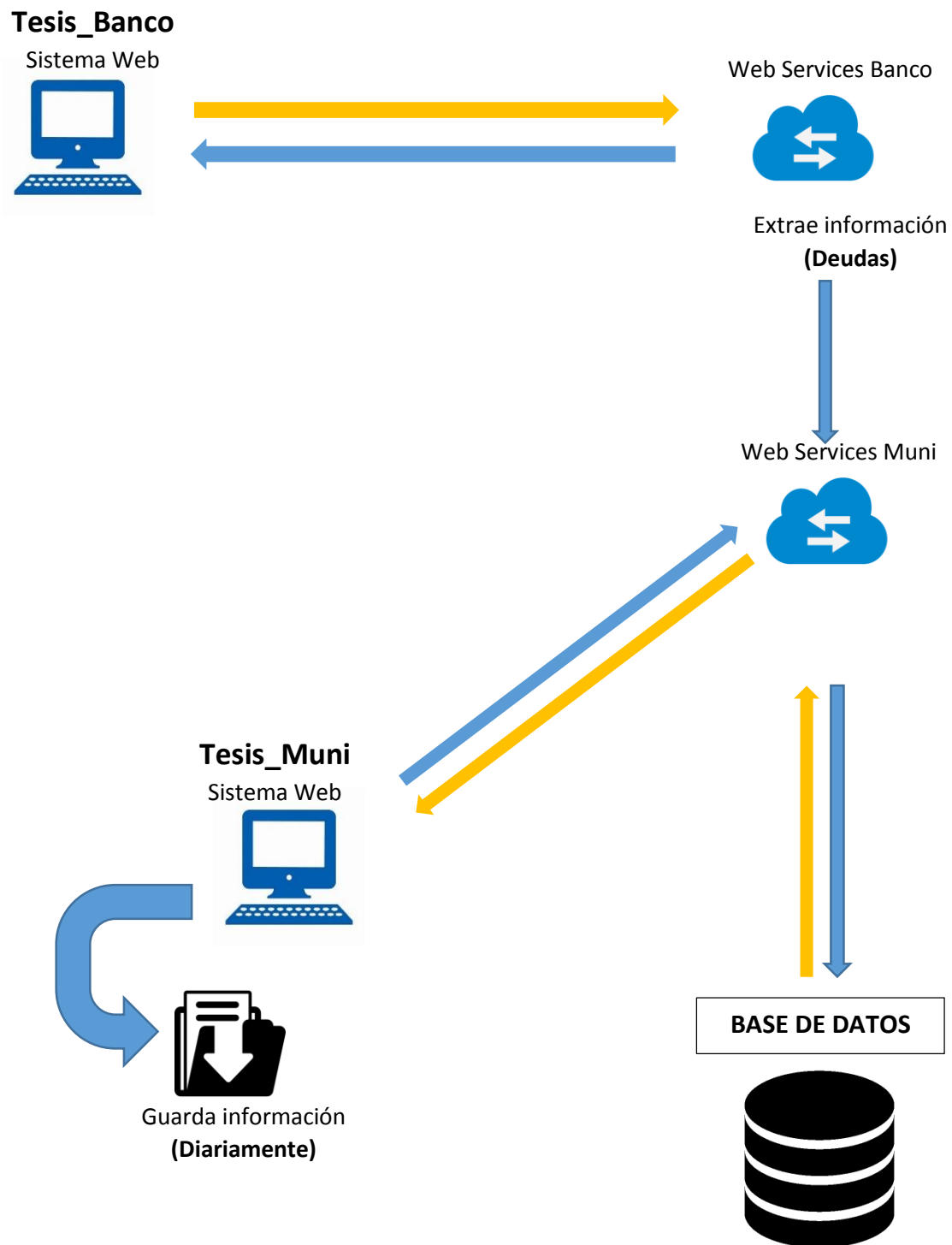


Figura 7: Diseño arquitectónico  
Autor: Elaboración propia



Nuestra aplicación Android y nuestro sistema web, realizan consultas mediante el protocolo HTTP hacia nuestra web services, mediante la API donde se encuentran los recursos que se van a consumir. Las peticiones HTTP que se están usando son GET o POST. La información devuelta se está retornando el formato JSON (application/json).

El servicio web identifica por medio de la URL el recurso que va a utilizar, además de los parámetros que podrían recibir si es que se da el caso.

Dicho recurso se conectará con la base de datos SQL Server y obtendrá la información solicitada.

La respuesta HTTP retornará a la aplicación para que se encargue de procesarla adecuadamente y manipularla para fines del desarrollo.

El otro sistema web de parte del banco, también se conectará a una web services y usa la misma estructura de consultas de petición y retornos de información.

La comunicación de las aplicaciones con nuestra web services se hace a través de REST. Para dicha comunicación se van a utilizar dos tipos de peticiones: GET (consulta y obtiene datos) y POST (crea recursos).

PETICIÓN	RECURSO	MÉTODOS
POST	login_muni/auth /por_cobrar /por_cobrar_filter /buscar_contribuyente /crear_credencial /enviar_cobros /pago_final	login_muni contribuyentes_por_cobrar contribuyentes_por_cobrar_filtro buscar_contribuyente crear_credencial enviar_cobros_contribuyentes pagos_finales
GET		

*Tabla 10: Peticiones POST / GET Municipalidad (Android)  
Autor: Relaboración propia*

PETICIÓN	RECURSO	MÉTODOS
POST	/login	auth
GET	/users /predio_predial /predio_all_predial /anio_deuda_x_contribuyente /cuota_x_anio_contribuyente /arbitrio_limpieza /arbitrios_limpieza_x_anios /cuotas_x_anio_arbitrios /pagos_x_contribuyente	informacion_usuario predio_predial predio_all_predial anio_deuda_contribuyente cuota_x_anio obtener_arbitrio_limpieza_publica arbitrios_l_x_anios_deuda_contribuyente cuotas_x_anio_arbitrios pagos_x_contribuyente

Tabla 11: Peticiones POST / GET Municipalidad (Web)  
Autor: Elaboración propia

PETICIÓN	RECURSO	MÉTODOS
POST	/auth_banco /deudores /pagar_x_banco /no_deudores /enviar_pagos /hora_atencion	user_credencial_banco mostrar_clientes_deudores pagar_deuda_x_banco mostrar_clientes_no_deudores enviar_pago_desde_banco obtener_hora_atencion
GET		

Tabla 12: Peticiones POST / GET Banco (Web)  
Autor: Elaboración propia

Cada una de estas URLs tienen una funcionalidad en la Web Services, es por ello que estarán asignadas a distintos métodos de nuestras clases. Comentaremos que hace cada método:

**[login\_muni]:** Controla el acceso al sistema web de la municipalidad, recibiendo un usuario y contraseña.

**[contribuyentes\_por\_cobrar]:** Retorna un objeto JSON trayendo toda la data de los contribuyentes que adeudan.

**[contribuyentes\_por\_cobrar\_filtro]:** Retorna un objeto JSON trayendo toda la data de los contribuyentes que adeudan. Permitiendo recibir el mes y el año para hacer filtros de búsqueda.

**[buscar\_contribuyente]:** Retorna los datos de los contribuyentes en formato JSON, te permite la entrada de un parámetro para realizar dicha búsqueda.

**[crear\_credencial]:** Permite crear credenciales para los usuarios del sistema.

**[enviar\_cobros\_contribuyentes]:** Envía todos los contribuyentes deudores al banco.

**[pagos\_finales]:** Realiza una transacción de los contribuyentes que ya han pagado al banco.

**[auth]:** Controla el acceso del aplicativo móvil de la municipalidad, recibiendo un usuario y contraseña.

**[predio\_predial]:** Retorna todos los predios de contribuyentes.

**[predio\_all\_predial]:** Retorna los predios de forma global de los contribuyentes.

**[anio\_deuda\_contribuyente]:** Lista las deudas de los contribuyentes por año.

**[enviar\_pago\_desde\_banco]:** Realiza una transacción de los contribuyentes que ya han pagado al banco.

### 5.3.2. PETICIONES GET

Estableceremos un contexto para entender cómo funciona un API por el método GET.

Describiremos el proceso de listado de predios, lo primero que se debe comprobar es el token y posteriormente se realizará una petición GET/predio\_predial a nuestro servicio web e incluyendo el campo requerido. El parámetro requerido:

- ✓ Headers[token]
- ✓ codCont
- ✓ anio

El servicio web detectará la URL, que se encuentra apuntando al controlador con su respectivo método “**predio\_predial\_get()**”. Tareas del método:

- Este método detectará que el usuario genere un token.
- Posteriormente hará una consulta hacia el modelo del proyecto que este se conectará hacia la base de datos.
- La base de datos recibirá los datos del API y ejecutará la consulta respectiva.
- El API recibirá los datos de dicha consulta y retornará toda la información en formato JSON.

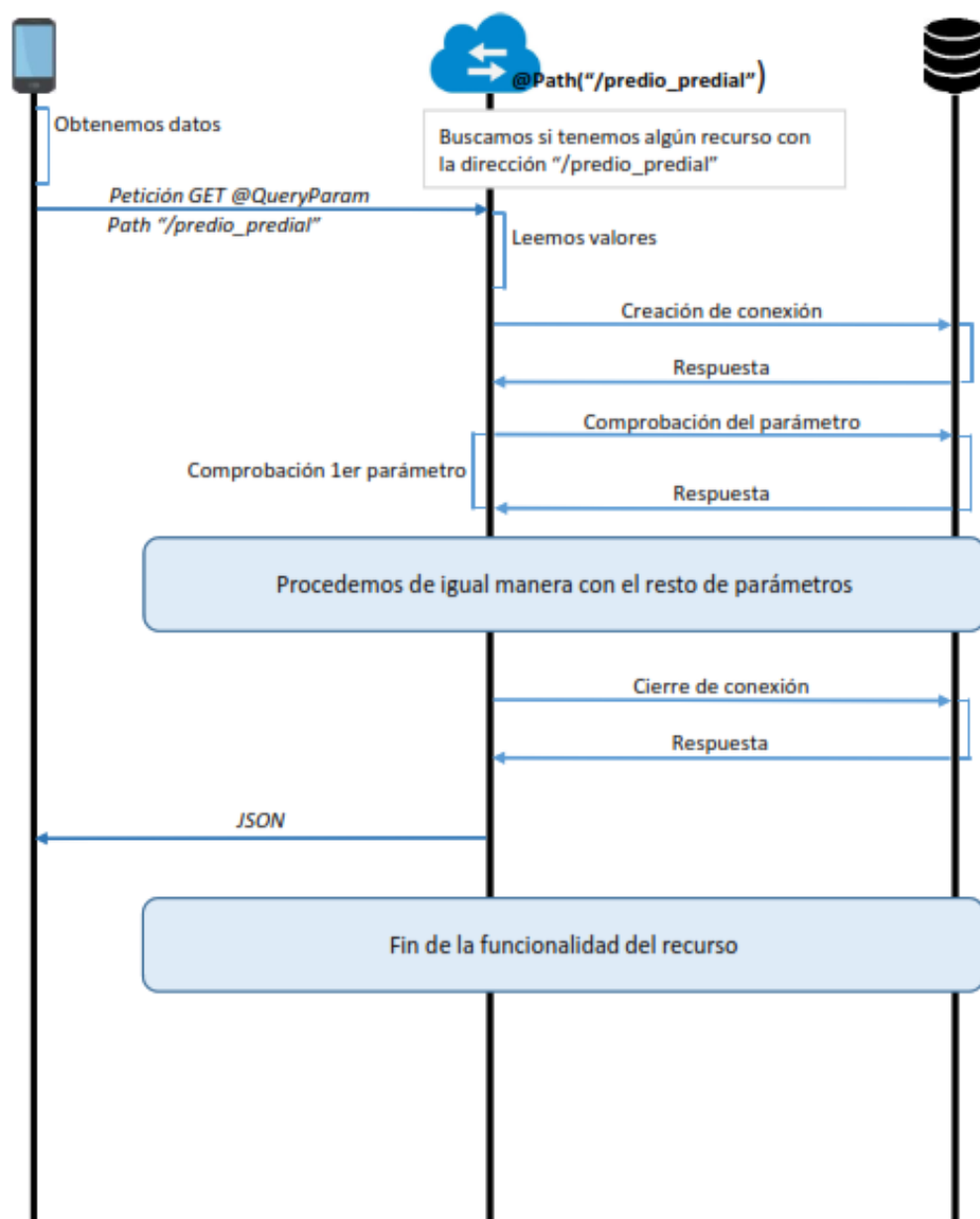


Figura 8: Petición GET  
Autor: Elaboración propia

### 5.3.3. PETICIONES POST

Estableceremos un contexto para entender cómo funciona un API por el método POST. Describiremos el proceso de autenticación móvil, se realizará una petición POST/login a nuestro servicio web e incluyendo los campos requeridos. Los parámetros requeridos son:

- ✓ Username
- ✓ Password

El servicio web detectará la URL que se encuentra apuntando al controlador con su respectivo método **“auth\_post()”**. Tareas del método:

- Este método detectará que el usuario ingresado exista, en este caso hará la consulta al modelo que se conectará hacia la base de datos.
- La base de datos recibirá el usuario y ejecutará la consulta respectiva.
- La consulta retornará los datos del usuario, en esos datos retornará la contraseña y se evaluará si es la misma contraseña ingresada.
- Luego se codificará el token con todos los datos retornado de la consulta del usuario.
- El API retornará toda la información en dicho token en formato JSON.

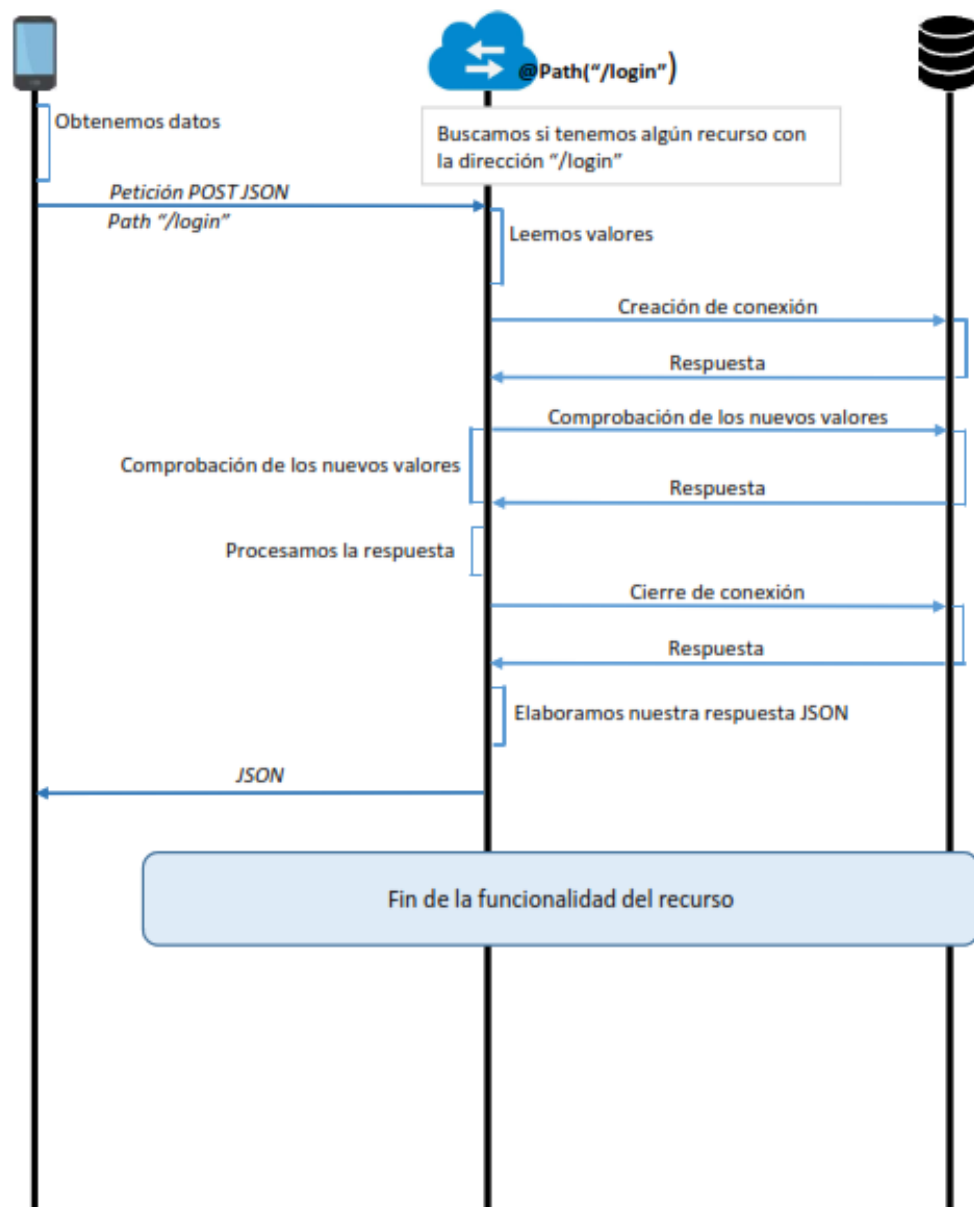
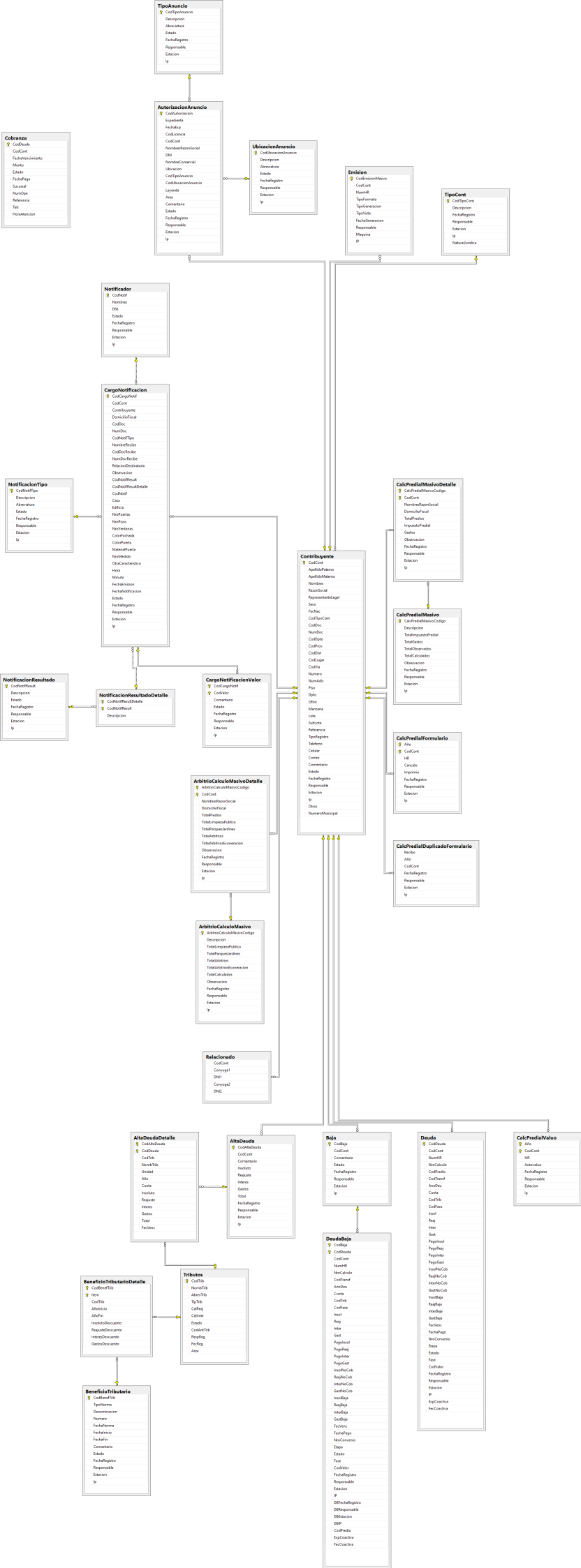


Figura 9: Petición POST  
Autor: Elaboración propia



### 5.3.5. CONFIGURACIÓN DEL SERVIDOR AZURE

#### a. Creación del Servidor y de la base de datos.

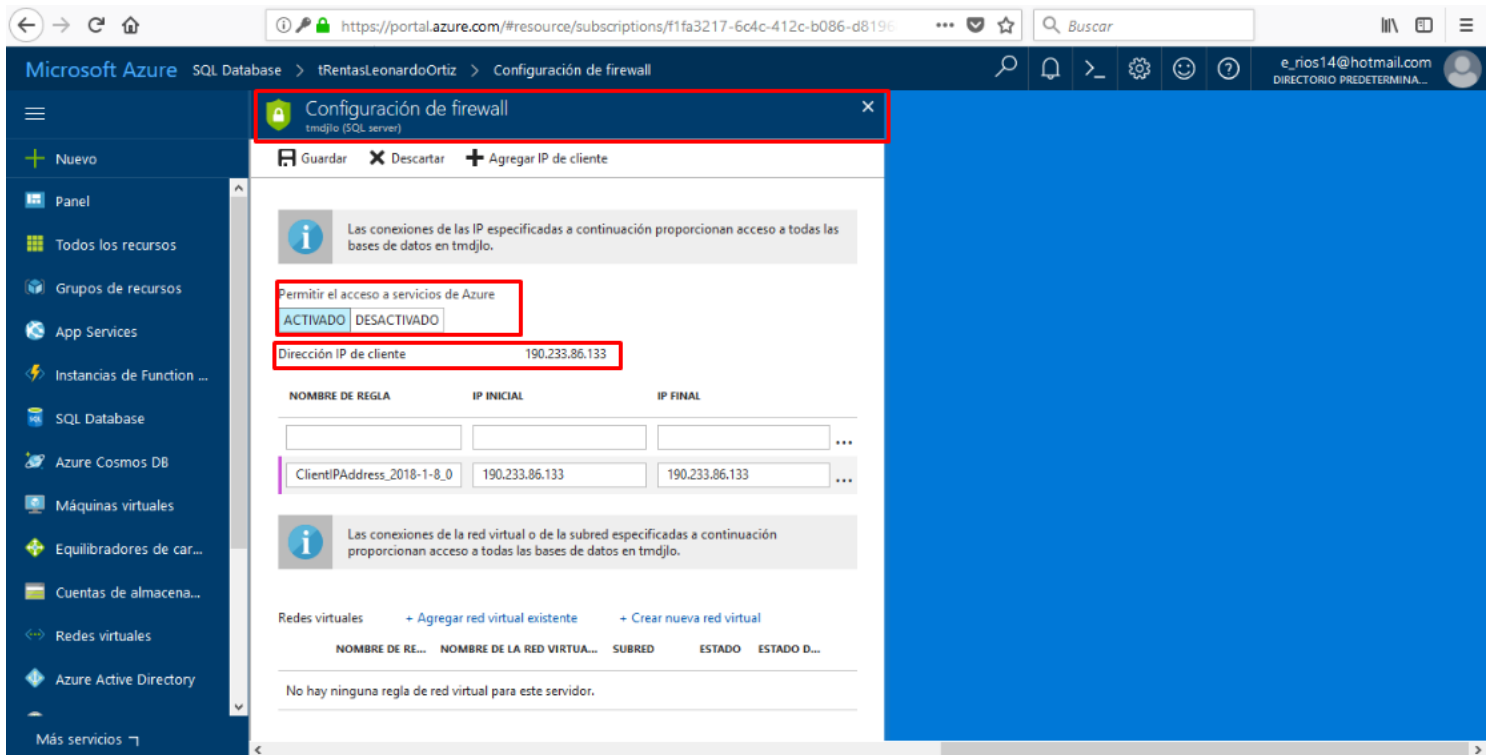
Captura de pantalla 1: Creación del servidor y de la base de datos

#### b. Información de la base de datos y características.

Captura de pantalla 2: Información de la base de datos y características

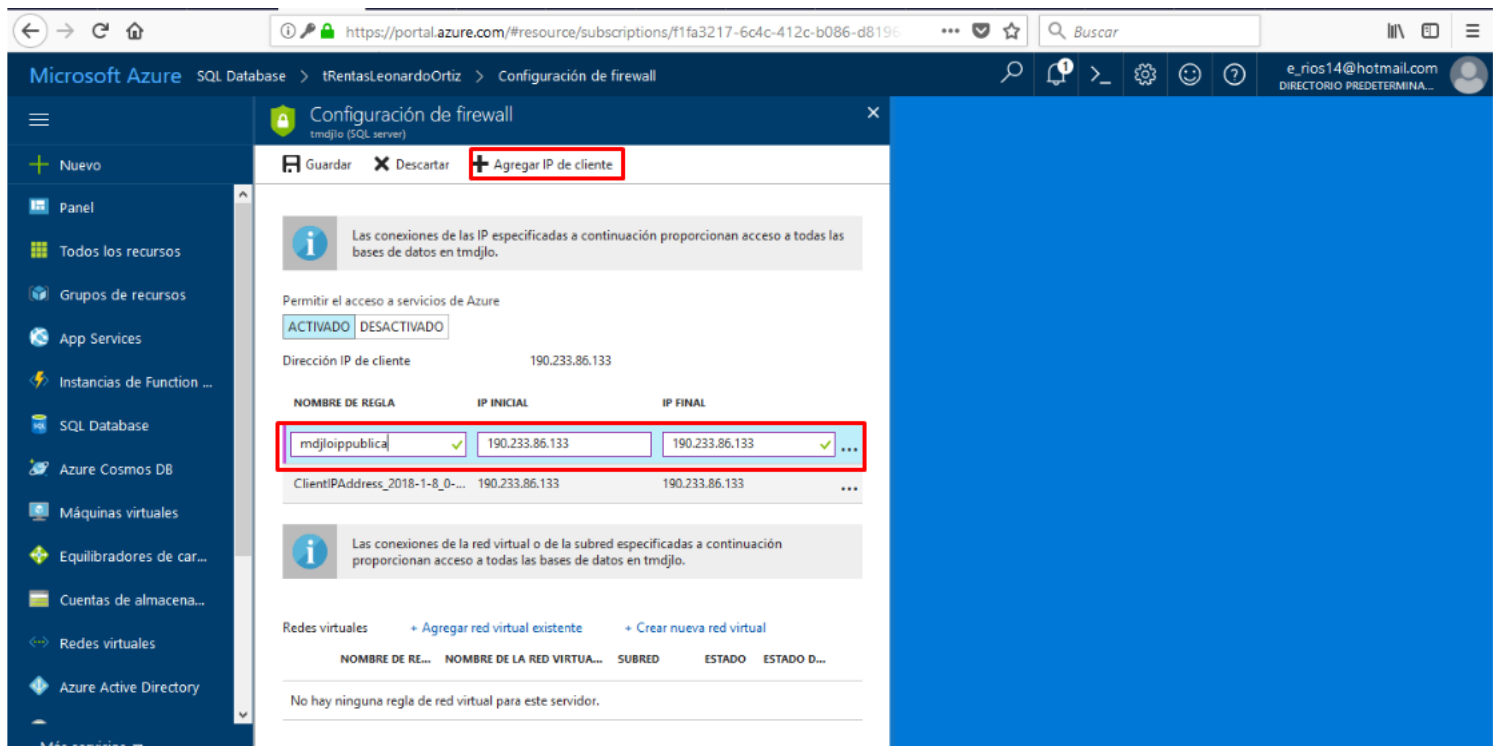


### c. Configuración de firewall.



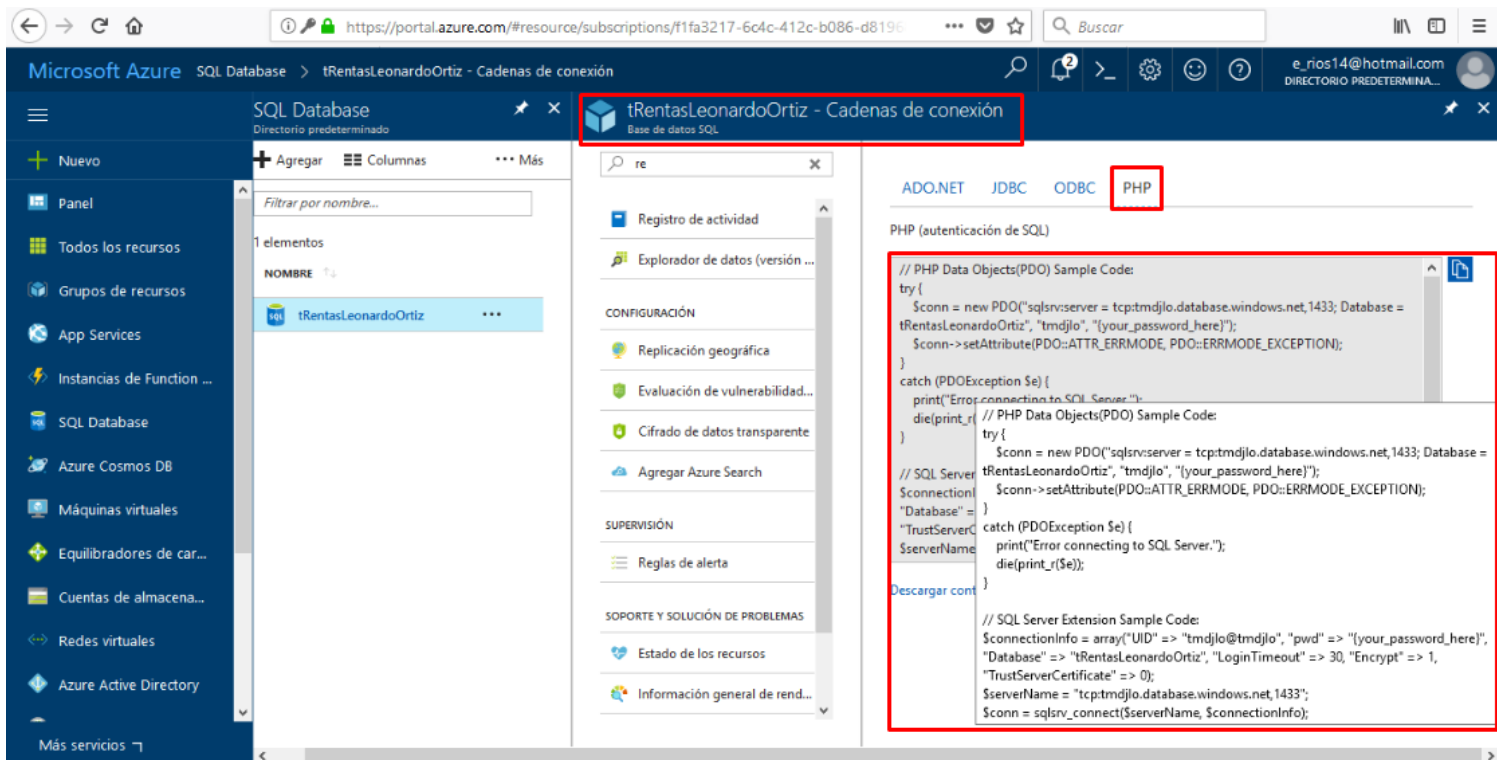
Captura de pantalla 3: Configuración de firewall

### d. Agregando IP del Cliente



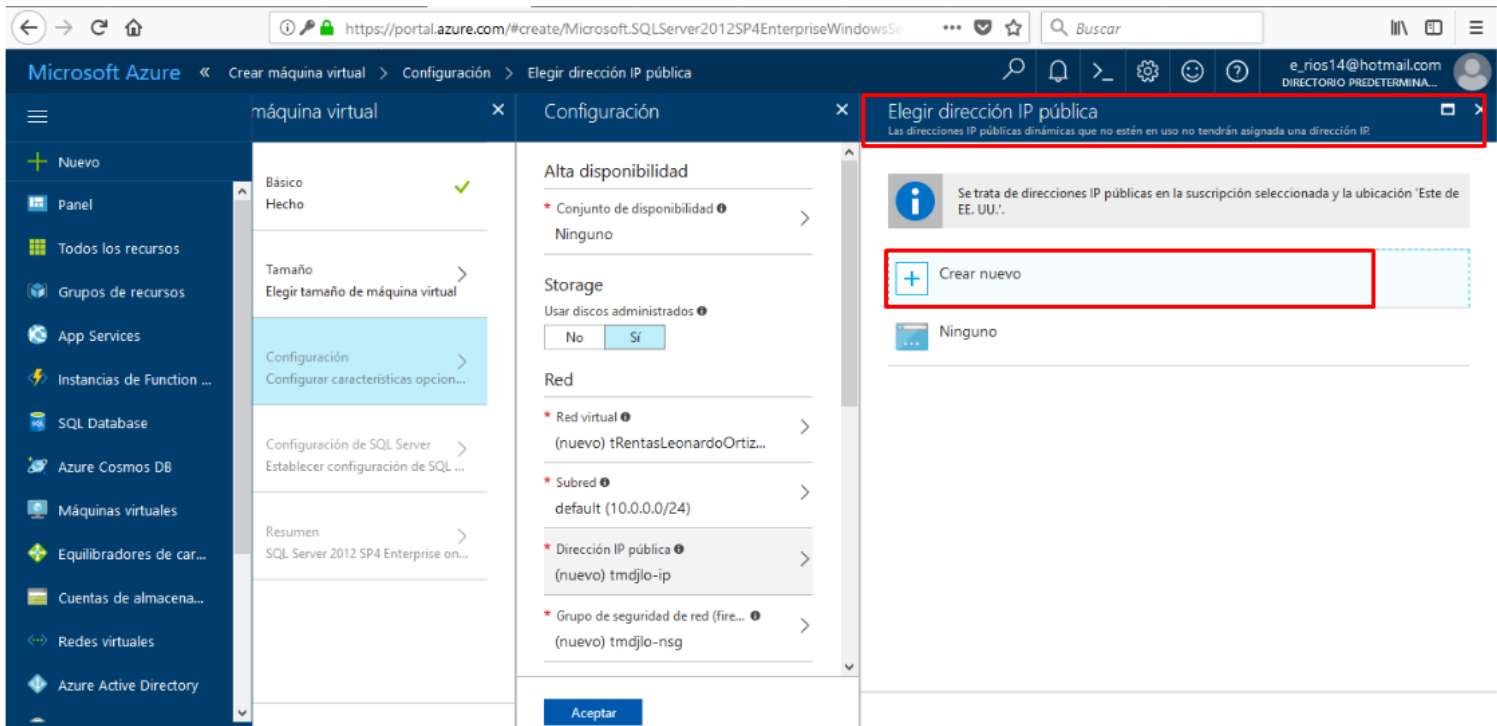
Captura de pantalla 4: Agregando IP del cliente

## e. Cadena de Conexión.



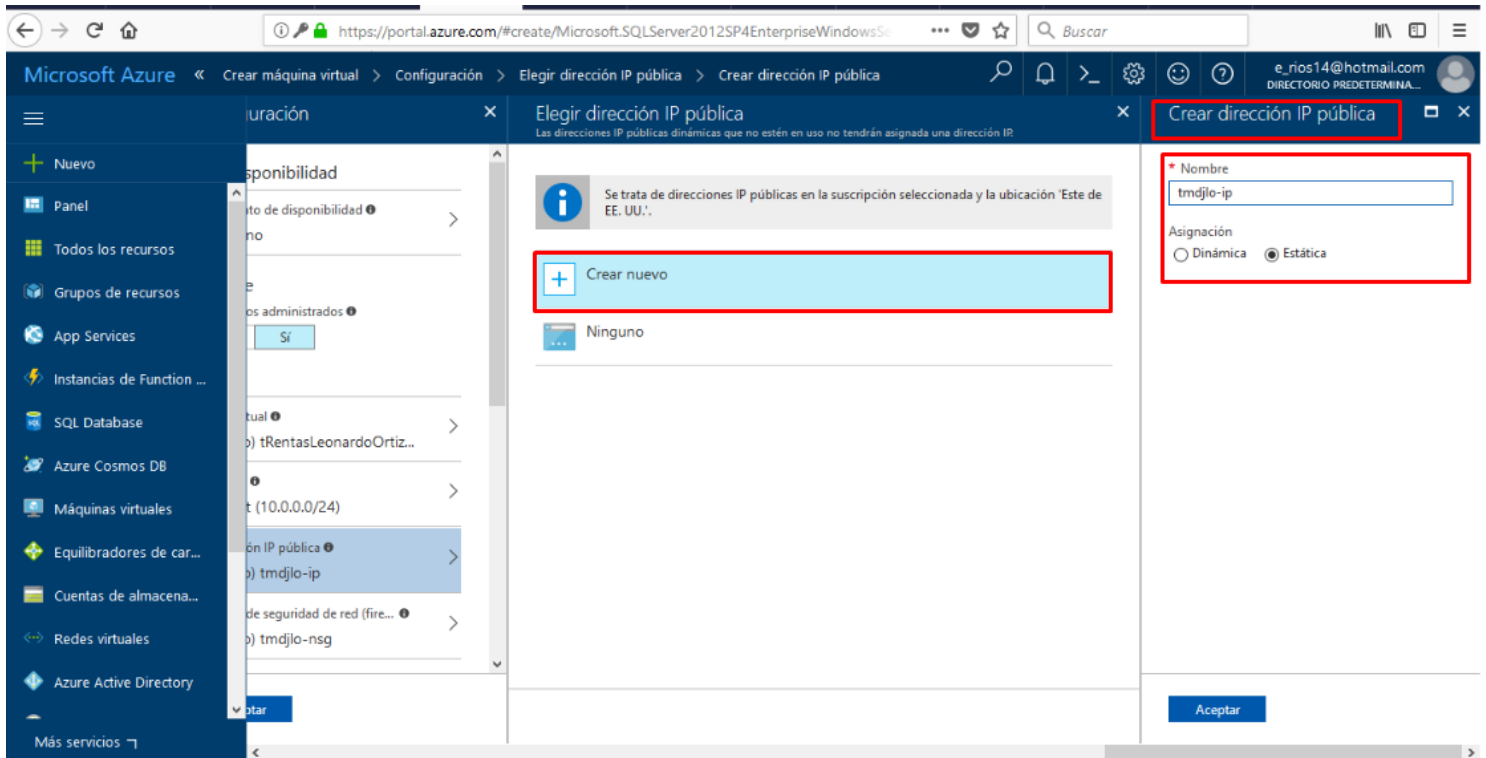
Captura de pantalla 5: Cadena de conexión

## f. Elegir Dirección IP Pública.



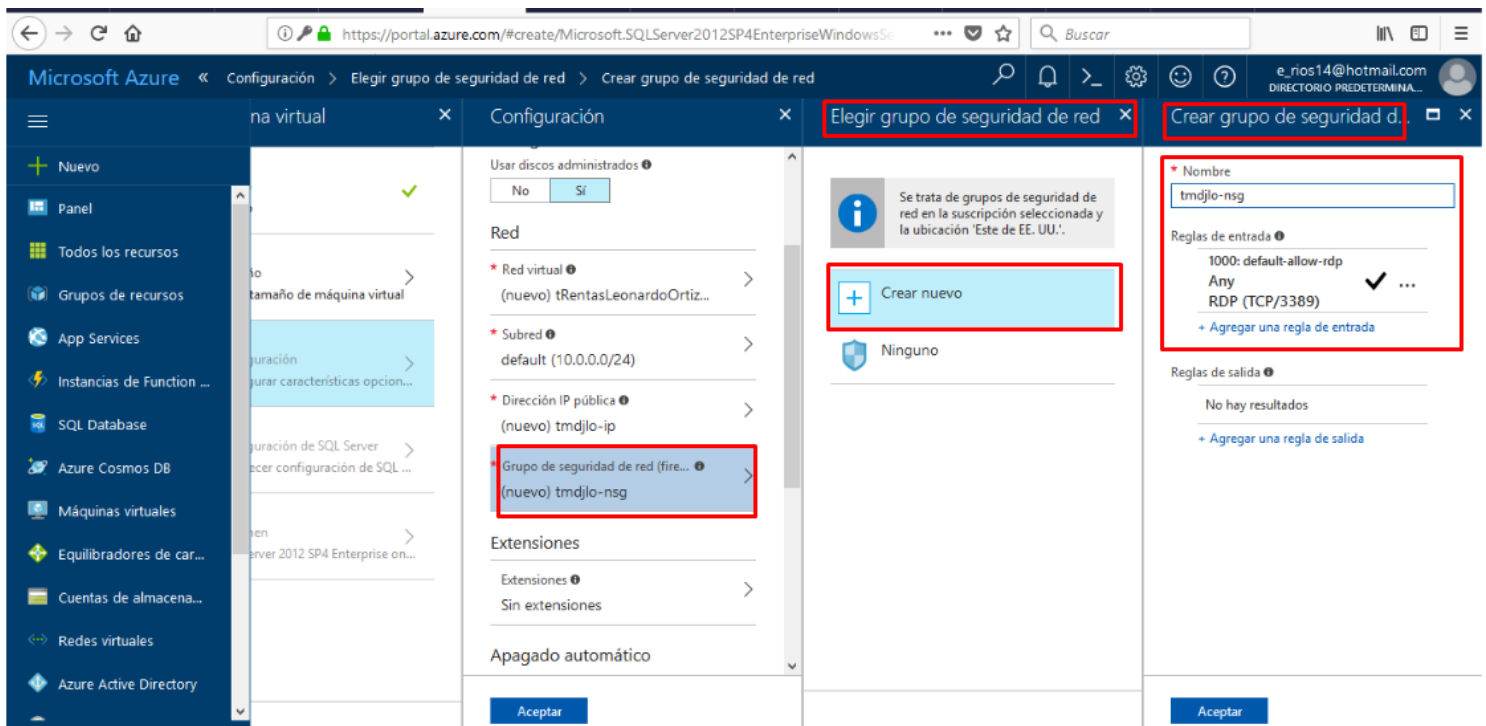
Captura de pantalla 6: Elegir dirección IP pública

## g. Creación Dirección IP Pública.



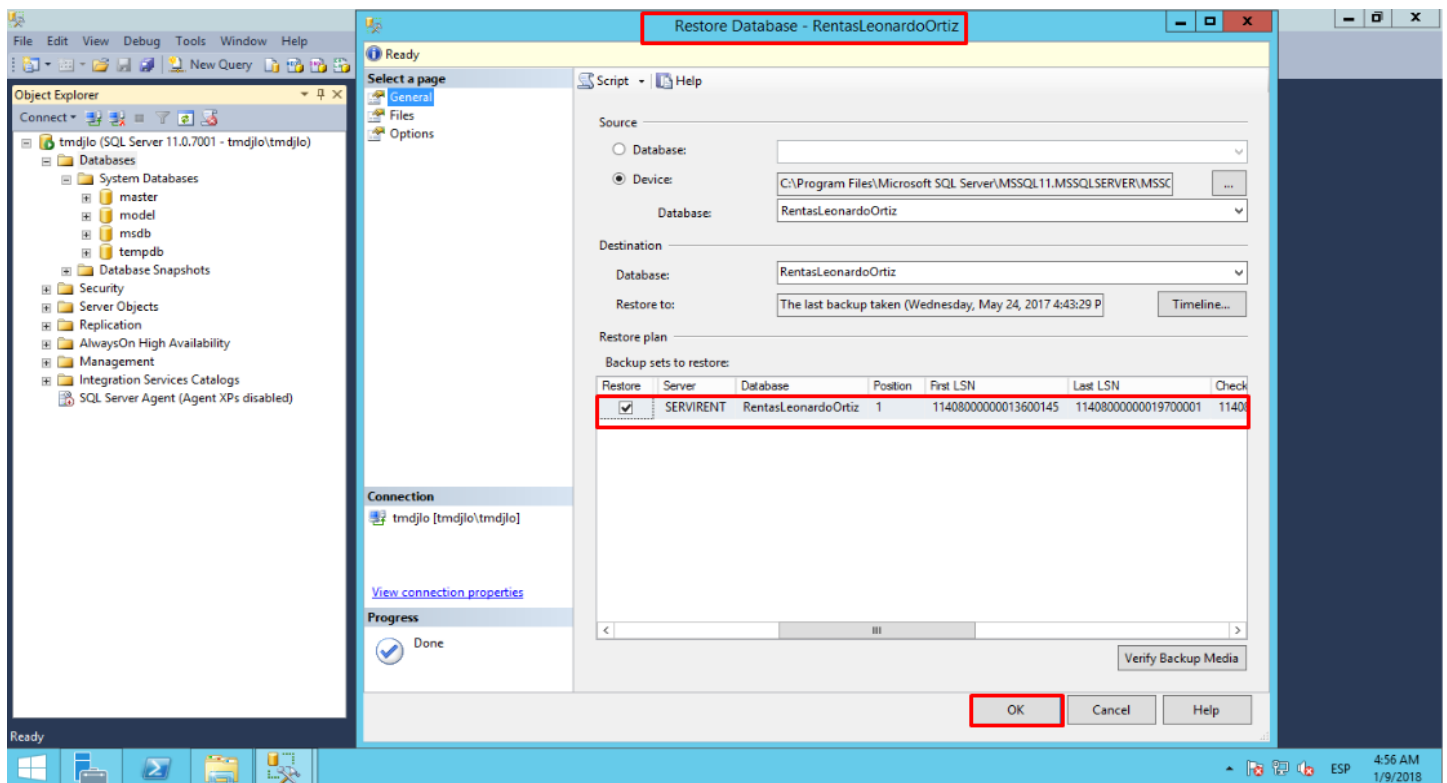
Captura de pantalla 7: Creación dirección IP pública

## h. Elegir y Crear Grupo de Seguridad de Red.



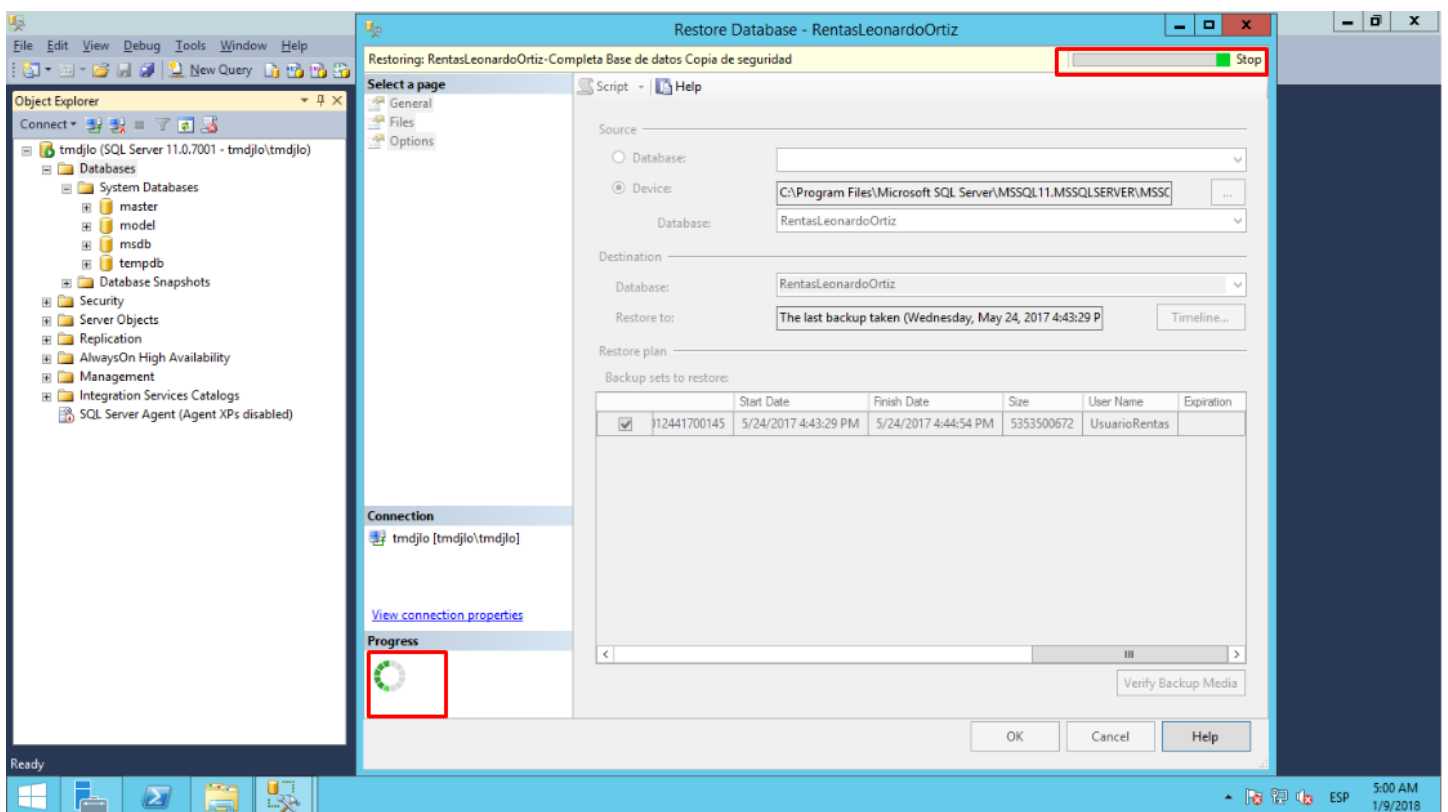
Captura de pantalla 8: Elegir y crear grupo de seguridad de red

## i. Máquina virtual y restauración de la Base de Datos.



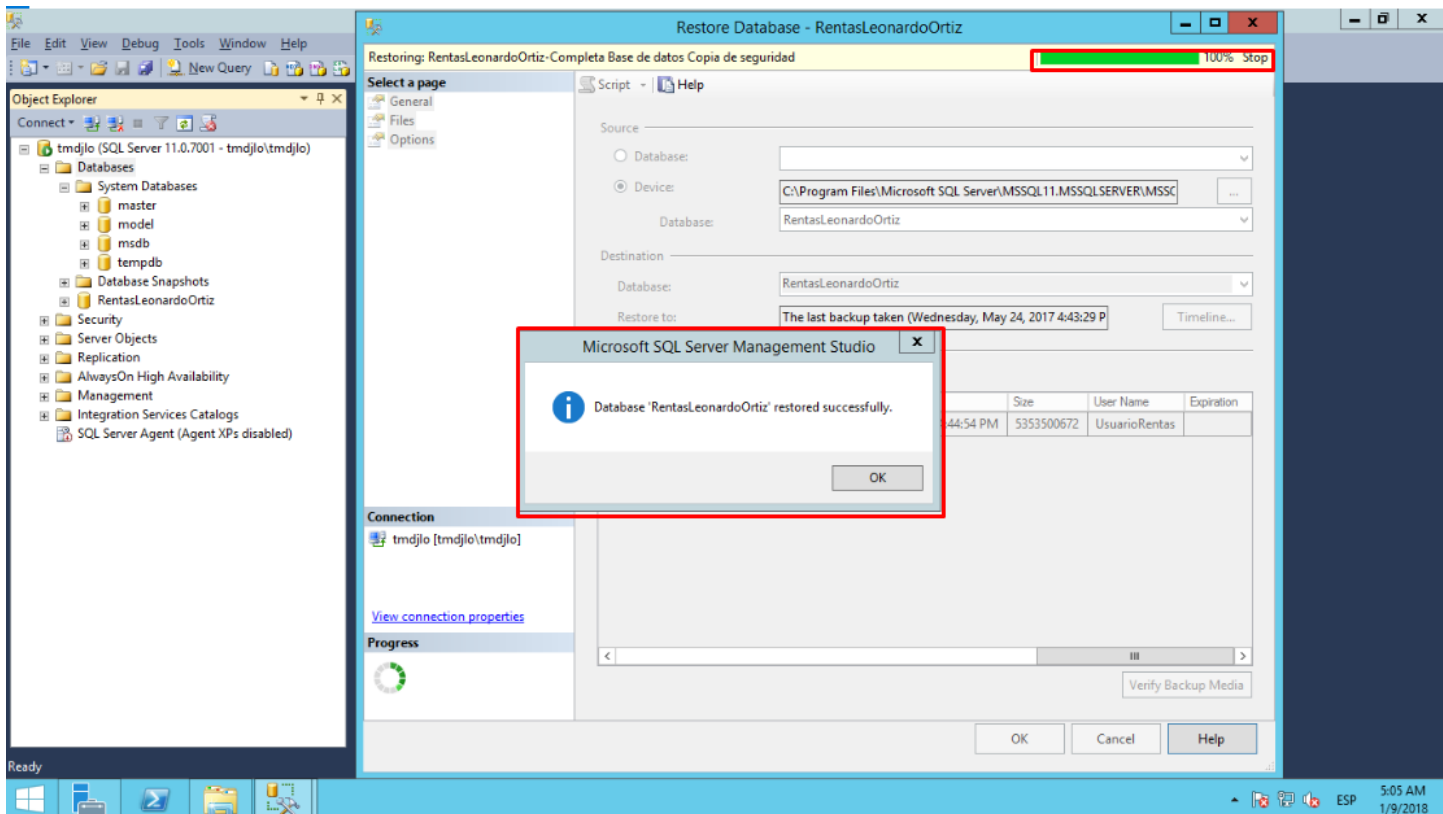
Captura de pantalla 9: Máquina virtual y restauración de la base de datos

## j. Iniciación de Restauración de la Base de Datos.



Captura de pantalla 10: Iniciación de restauración de la base de datos

k. Finalización de restauración de la Base de Datos.



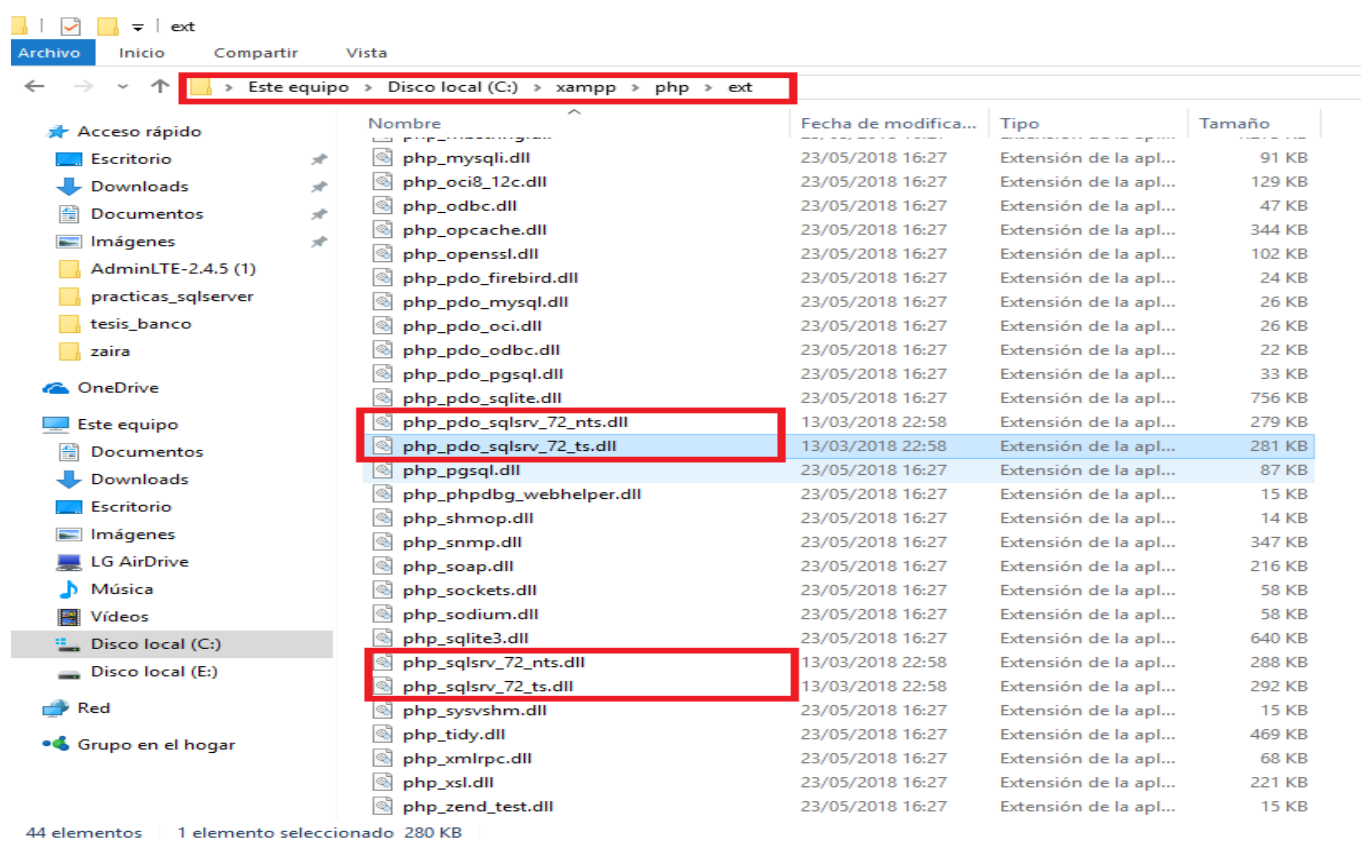
Captura de pantalla 11: Finalización de restauración de la base de datos

### 5.3.6. CONFIGURACIÓN DE HERRAMIENTAS Y DRIVERS

En la imagen que se muestra los archivos .dll que nos brindan la comunicación con SQL Server.

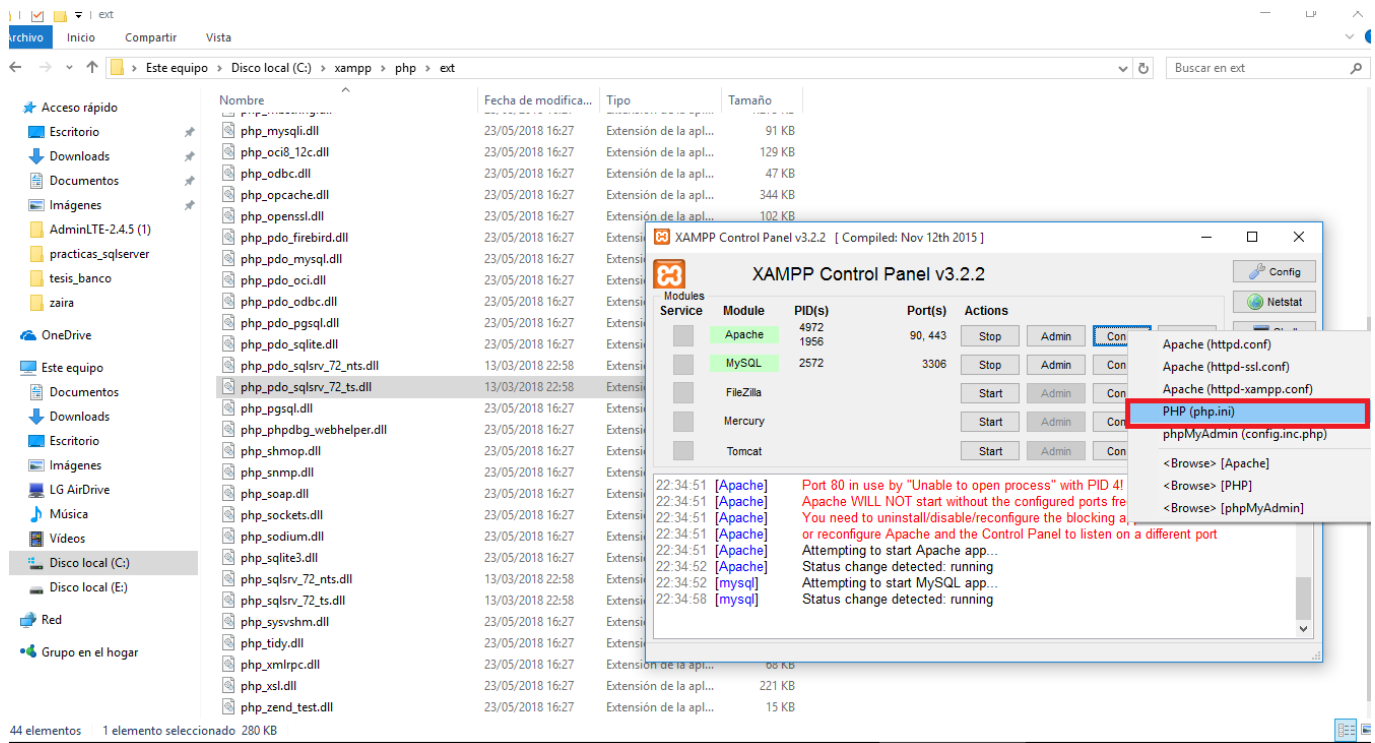
- ✓ php\_pdo\_sqlsrv\_72\_nts.dll
- ✓ php\_pdo\_sqlsrv\_72\_ts.dll

Estos drivers tienen compatibilidad con PHP versión 7.2. En la imagen también se muestra donde se encuentran localizados para poder hacer uso de ellos.



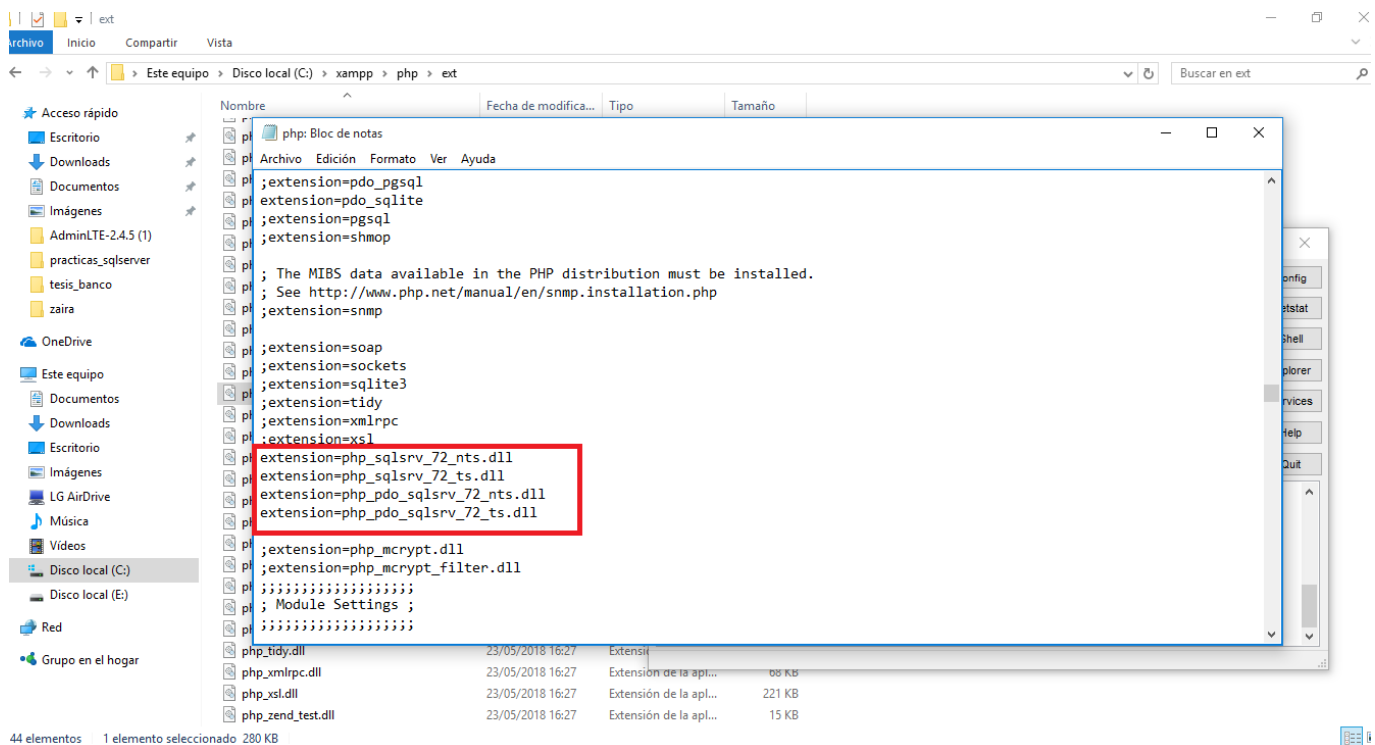
Captura de pantalla 12: Agregando los archivos .dll para el funcionamiento SQL Server

## Abrimos el archivo php.ini del servidor xampp



Captura de pantalla 13: Abriendo carpeta para la configuración de las extensiones del driver SQL Server

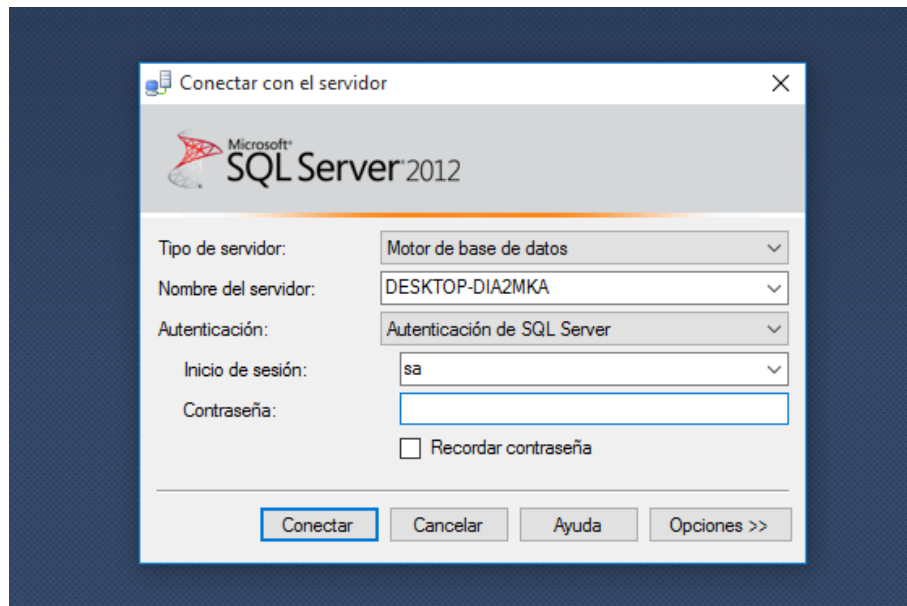
Luego tenemos que configurar los archivos .dll en el php.ini del servidor xampp.



Captura de pantalla 14: Configurando las extensiones SQL Server del xampp



### 5.3.7. CONFIGURACIÓN DE USUARIOS DE SQL SERVER

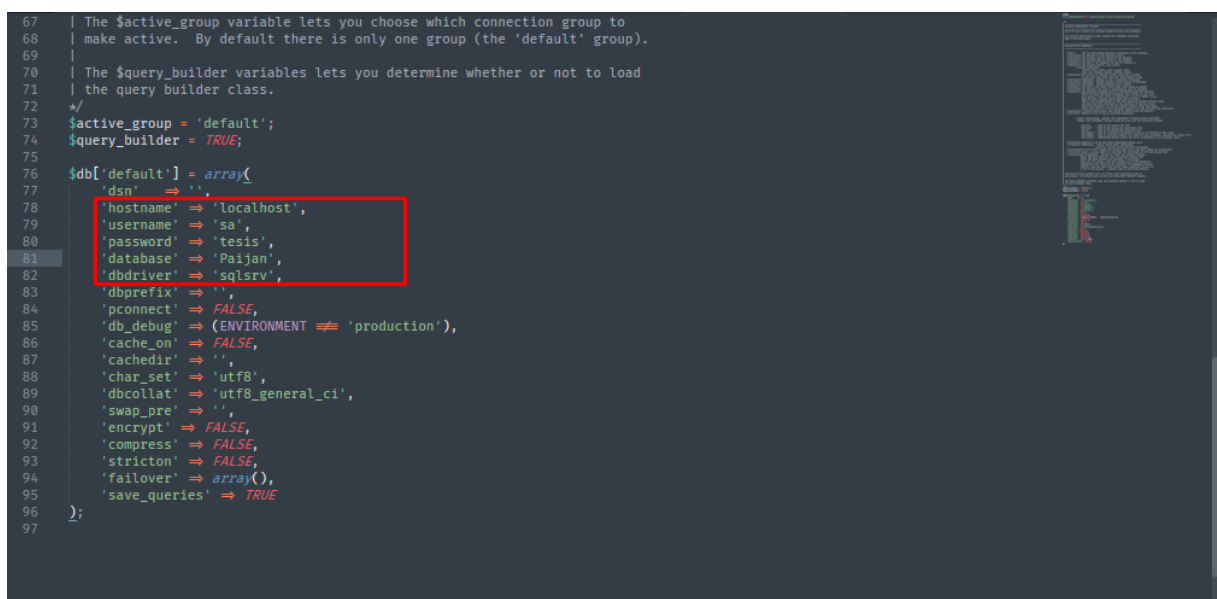


Captura de pantalla 15: Iniciando sesión de la base de datos SQL Server

### 5.3.8. CONFIGURACIÓN DE LA WEB SERVICES

En el archivo database.php realizamos la configuración de la conexión a la base de datos SQL Server ingresando:

- ✓ **username:** sa
- ✓ **password:** tesis
- ✓ **database:** Paijan
- ✓ **driver:** sqlsrv



Captura de pantalla 16: Configuración de la conexión a la base de datos de la Web Services



En el archivo rest.php nos permite configurar un key y la expiración del token y el formato del algoritmo HS256.

```

106 | FALSE      No login required
107 | 'basic'     Unsecured login
108 | 'digest'    More secured login
109 | 'session'   Check for a PHP session variable. See 'auth_source' to set the
110 |             authorization key
111 | 'jwt'       Check jwt bearer on authorization header and validate it.
112 |
113 | */
114 | $config['rest_auth'] = false;
115 |
116 | /*
117 | -----
118 | JWT Settings
119 | -----
120 |
121 | Required for jwt authentication
122 |
123 | */
124 |
125 | $config['jwt_secret_key'] = 'ingDLmRuGe9UKHRNjs7cYckS2yu14lc3';
126 |
127 | $config['jwt_token_expire'] = 900; //900 -- 900 segundos entre 60 segundos = 15 min
128 |
129 | $config['jwt_algorithm'] = 'HS256';
130 |
131 | /*
132 | -----
133 | REST Login Source
134 | -----
135 |
136 | Is login required and if so, the user store to use
137 |
138 | ''         Use config based users or wildcard testing
139 | 'ldap'     Use LDAP authentication
140 | 'library'  Use a authentication library
141 |
142 | Note: If 'rest_auth' is set to 'session' then change 'auth_source' to the name of the session variable
143 |
144 | */
145 | $config['auth_source'] = '';
146 |

```

Captura de pantalla 17: Configuración del tiempo de expiración de los usuarios del sistema, claves secretas

En el archivo config.php declaramos la ruta base de nuestra web services [https://dominio.com/wb\\_tesis](https://dominio.com/wb_tesis), en este caso sería: [http://192.168.1.8:90/wb\\_tesis/](http://192.168.1.8:90/wb_tesis/)

```

1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 /*
5 |-----
6 | Base Site URL
7 |-----
8 |
9 | URL to your CodeIgniter root. Typically this will be your base URL,
10 | WITH a trailing slash:
11 |
12 | http://example.com/
13 |
14 | WARNING: You MUST set this value!
15 |
16 | If it is not set, then CodeIgniter will try guess the protocol and path
17 | your installation, but due to security concerns the hostname will be set
18 | to $_SERVER['SERVER_ADDR'] if available, or localhost otherwise.
19 | The auto-detection mechanism exists only for convenience during
20 | development and MUST NOT be used in production!
21 |
22 | If you need to allow multiple domains, remember that this file is still
23 | a PHP script and you can easily do that on your own.
24 |
25 | */
26 | $config['base_url'] = 'http://192.168.1.8:90/wb_tesis/';
27 |
28 | /*
29 | -----
30 | Index File
31 | -----
32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 | */
38 | $config['index_page'] = 'index.php';
39 |
40 | /*
41 | -----
42 | URI PROTOCOL
43 | -----

```

Captura de pantalla 18: Configuración de la ruta base para acceder a la web services

En el archivo routers.php configuramos nuestras endpoints para nuestro servicio web

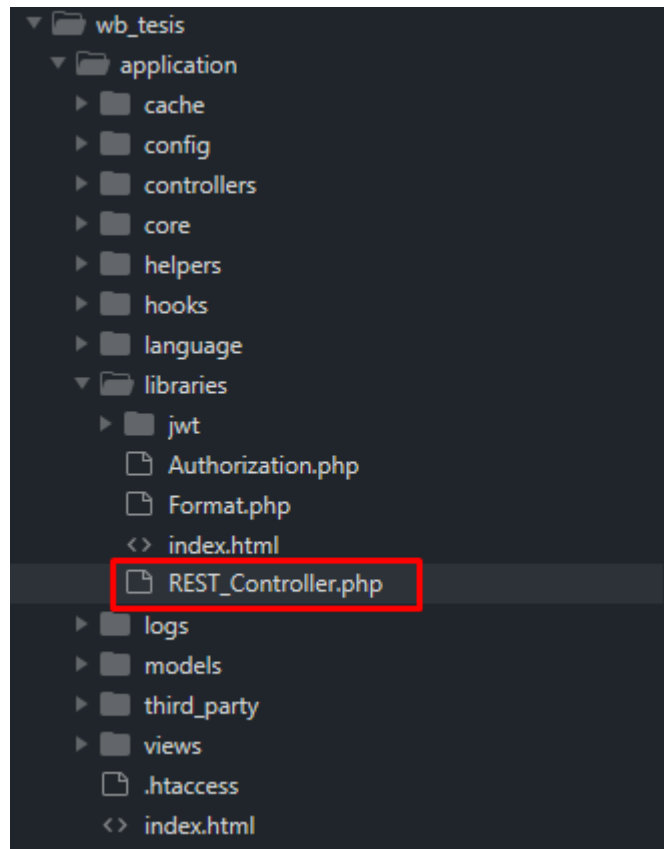
```

52 $route['default_controller'] = 'auth';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
55
56
57 //routes(controller/verbo) = controller/method
58
59 //web
60 $route['login_muni/auth']['post'] = 'auth/login_muni';
61 $route['por_cobrar']['post'] = 'contribuyentes/contribuyentes_por_cobrar';
62 $route['por_cobrar_filtro']['post'] = 'contribuyentes/contribuyentes_por_cobrar_filtro';
63 $route['buscar_contribuyente']['post'] = 'contribuyentes/buscar_contribuyente';
64 $route['crear_credencial']['post'] = 'contribuyentes/crear_credencial';
65 $route['enviar_cobros']['post'] = 'contribuyentes/enviar_cobros_contribuyentes';
66 $route['pago_final']['post'] = 'contribuyentes/pagos_finales';
67
68
69 //authapp
70 $route['login']['post'] = 'authapp/auth';
71 $route['users']['get'] = 'authapp/informacion_usuario';
72 $route['predio_predial']['get'] = 'authapp/predio_predial';
73 $route['predio_all_predial']['get'] = 'authapp/predio_all_predial';
74 $route['anio_deuda_x_contribuyente']['get'] = 'authapp/anio_deuda_contribuyente';
75 $route['cuota_x_anio_contribuyente']['get'] = 'authapp/cuota_x_anio';
76
77 /* arbitrios */
78 $route['arbitrio_limpieza']['post'] = 'authapp/obtener_arbitrio_limpieza_publica';
79 $route['arbitrios_limpieza_x_anios']['get'] = 'authapp/arbitrios_l_x_anios_deuda_contribuyente';
80 $route['cuotas_x_anio_arbitrios']['get'] = 'authapp/cuotas_x_anio_arbitrios';
81 $route['pagos_x_contribuyente']['get'] = 'authapp/pagos_x_contribuyente';
82
83

```

Captura de pantalla 19: Configuración de las rutas de accesos a la web services con las peticiones HTTP

Cargamos la librería Rest\_Controller que nos proporciona métodos para la salida de los datos en formato JSON y también nos brinda los códigos HTTP



Captura de pantalla 20: Agregando la librería JWT y REST\_Controller

Vamos a importar o llamar a nuestra clase Rest\_Controller en el controlador Auth

```

1  <?php defined('BASEPATH') OR exit('No direct script access allowed');
2  date_default_timezone_set('America/Mexico City');
3  require APPPATH . "/libraries/REST_Controller.php";
4
5  class Auth extends REST_Controller{
6      public function __construct(){
7          parent::__construct();
8          $this->load->model('auth_model');
9      }
10     public function index_get(){
11         echo 'en el index auth';
12     }
13     /*
14     ----- WEB -----
15     */
16
17     public function login_muni_post()
18     {
19         $output_data = array();
20         $username= $this->input->post('username');
21         $password = $this->input->post('password');
22         if (!empty($username) and !empty($password)) {
23             $check = $this->auth_model->checkAuthentication($username, $password);
24             if ($check == false) {
25                 $output_data=array(
26                     'status' => 'error',
27                     'message' => 'credenciales del usuario incorrecta'
28                 );
29                 $this->response($output_data, REST_Controller::HTTP_OK);
30             }
31         }
32     }
33 }

```

Captura de pantalla 21: Configurando la librería REST\_Controller a nuestro controlador Auth

Codificación del token y su retorno en formato JSON

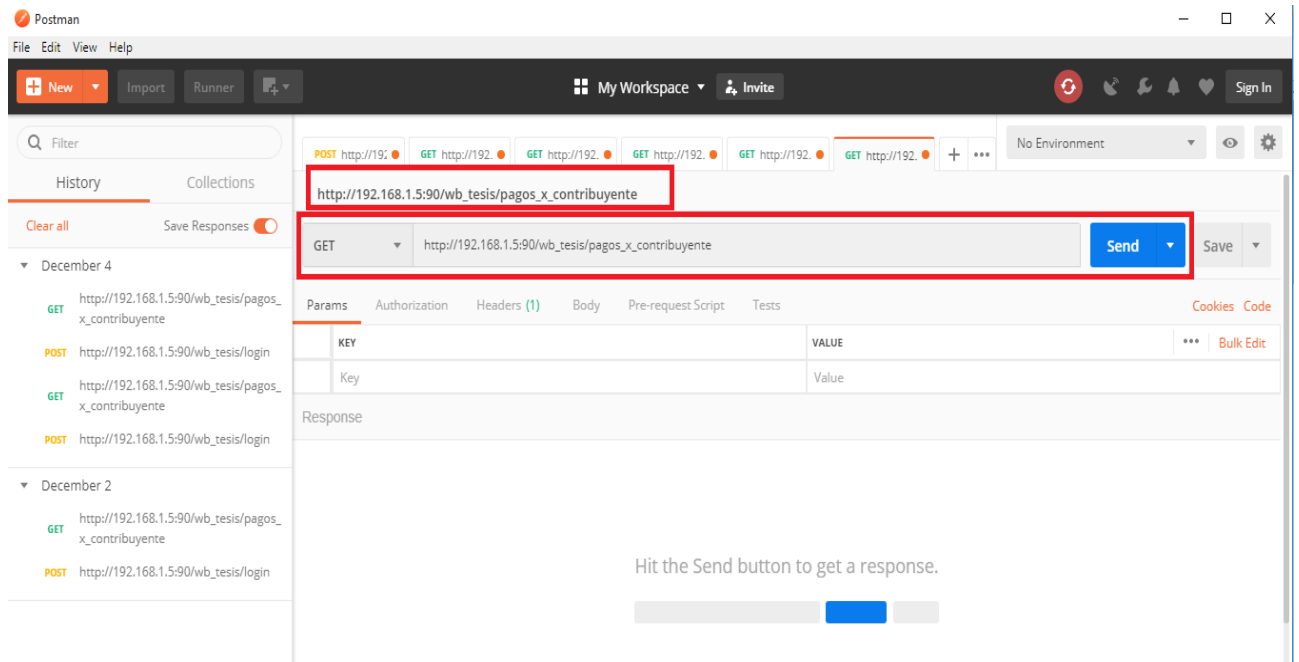
```

13     /*
14     ----- WEB -----
15     */
16
17     public function login_muni_post()
18     {
19         $output_data = array();
20         $username= $this->input->post('username');
21         $password = $this->input->post('password');
22         if (!empty($username) and !empty($password)) {
23             $check = $this->auth_model->checkAuthentication($username, $password);
24             if ($check == false) {
25                 $output_data=array(
26                     'status' => 'error',
27                     'message' => 'credenciales del usuario incorrecta'
28                 );
29                 $this->response($output_data, REST_Controller::HTTP_OK);
30             }
31             if ($check == true) {
32                 $date = new DateTime();
33                 $token = array(
34                     'iat' => $date->getTimestamp(), //tiempo de inicio del token
35                     'exp' => $date->getTimestamp() + $this->config->item('jwt_token_expire'), //tiempo que expira el token
36                     'username' => $check->username,
37                     'codCont' => $check->CodCont,
38                     'tipo' => $check->type_user
39                 );
40                 $output_data['status']='success';
41                 $output_data['message']= $this->jwt_encode($token); //codificamos el token
42                 $this->response($output_data, REST_Controller::HTTP_OK); // mostramos el token
43             }
44         }
45     }
46 }
47
48 //done

```

Captura de pantalla 22: Demostración de nuestra librería JWT, salida de datos

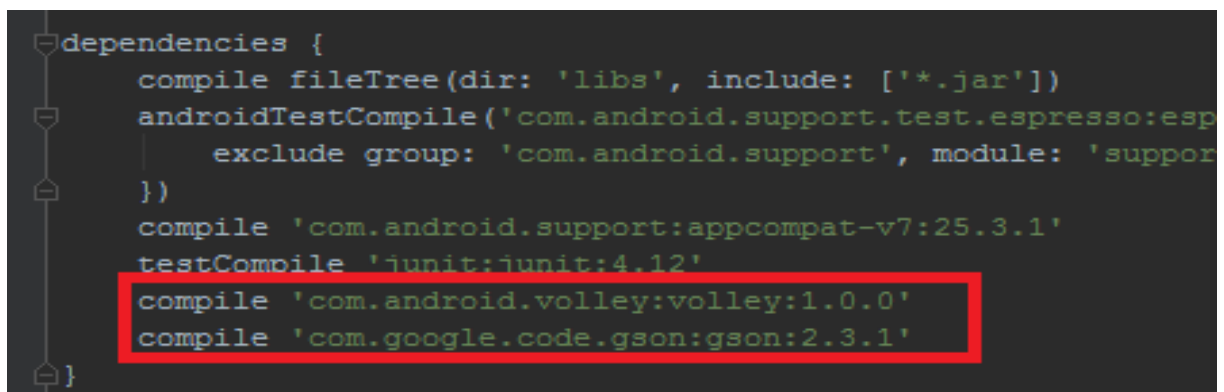
Utilizamos el POSTMAN para poder probar los endpoints de nuestra web service



Captura de pantalla 23: Realización de pruebas de datos de nuestra web services haciendo uso de POSTMAN

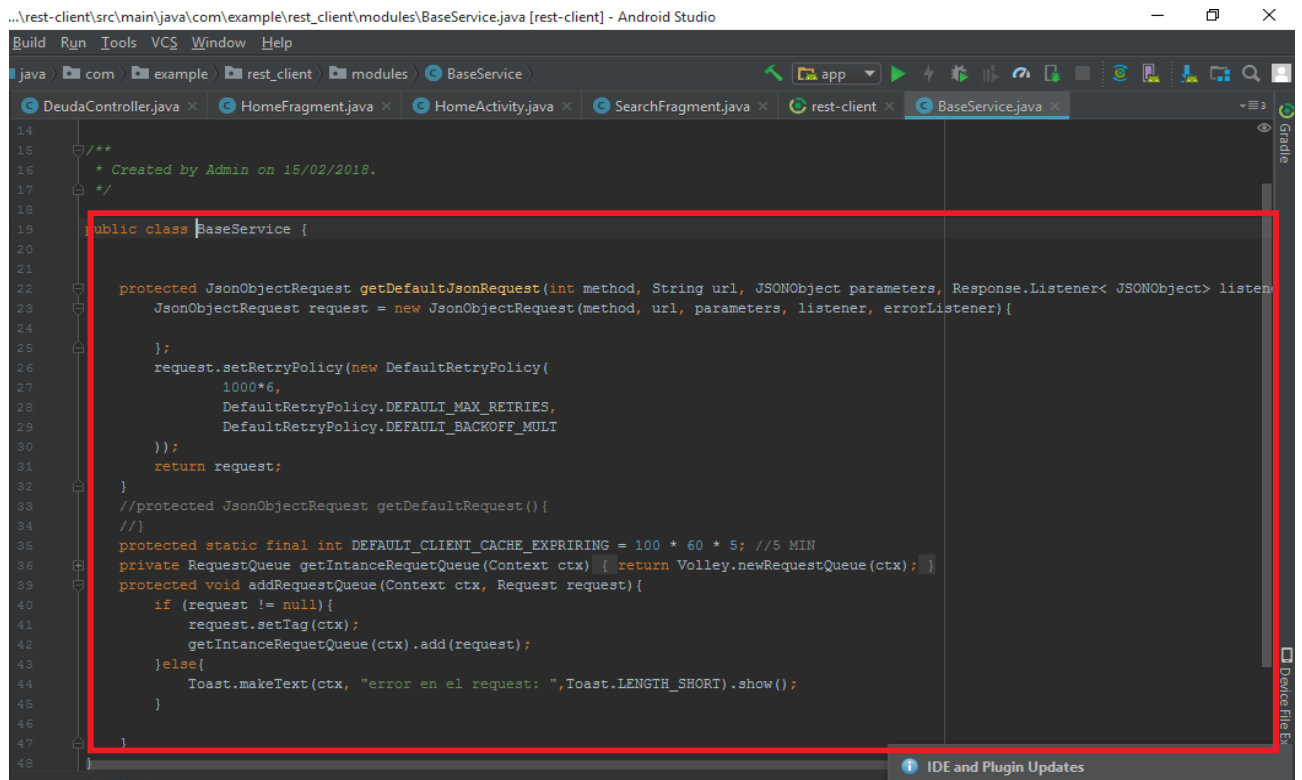
En la parte de la aplicación móvil, necesitaremos instalar dos librerías:

- ✓ **Volley:** Nos sirve para consumir los endpoints.
- ✓ **Gson:** Nos permite controlar nuestro código en formato JSON.



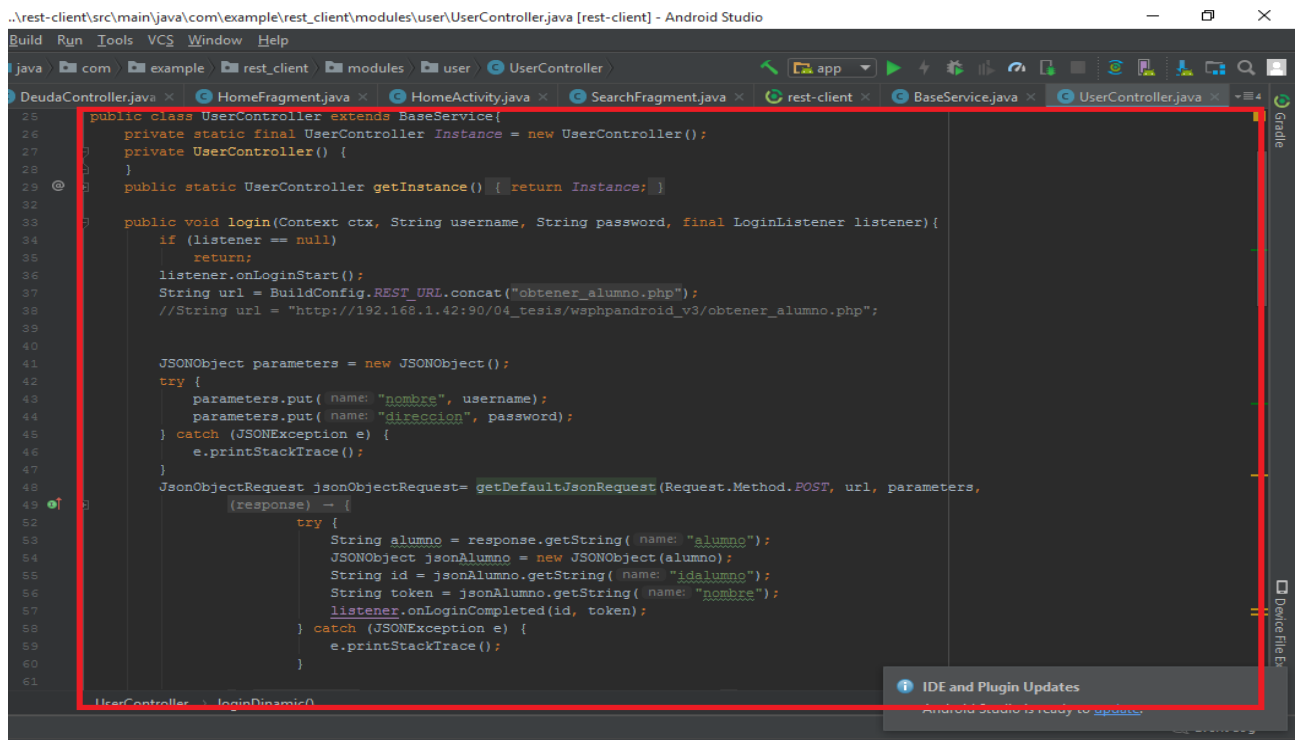
Captura de pantalla 24: Cargando la librería Volley y Gson para nuestra aplicación móvil

Luego, configuramos la librería Volley para la conexión de nuestra web services para posteriormente ser utilizada.



Captura de pantalla 25: Configuración de la librería Volley para la conexión a nuestra web services

Y posteriormente mostramos una parte del desarrollo de la aplicación móvil sobre la autenticación del usuario.

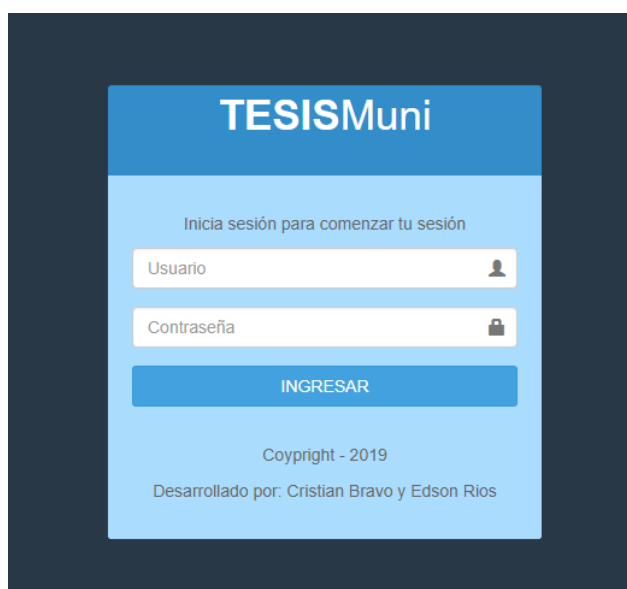


Captura de pantalla 26: Muestra del código de la autenticación de nuestra aplicación móvil

### 5.3.9. INTRODUCCIÓN AL ACCESO DEL SERVICIO

Explicaremos los sistemas web y la aplicación móvil, teniendo en cuenta de la intervención de todas las tecnologías.

En la parte web del sistema Tesis\_Muni accederemos con credenciales a través de un formulario.



Captura de pantalla 27: Inicio de sesión

Después de haber ingresado las credenciales correctas, el sistema nos mostrará un panel de control donde se muestra opciones como la creación de archivo, el envío de cobros de forma filtrado y global.



Captura de pantalla 28: Panel de control principal

### 5.3.9.1. ENVIAR COBROS FILTRADO

Se va a listar los contribuyentes que contengan deudas en el mes y año seleccionado.

TesisMuni		Administrador		
<div> Principal Gestión de Contribuyente Crear Credenciales Listado Credenciales Gestión de Cobranza Cerrar Sesión </div>		<div> Cobros Inicio &gt; Filtro </div>		
		<div> Seleccionar Mes Elegir... Seleccionar Año Elegir... Buscar </div>		
		<div> Enviar Deuda </div>		
COD. Contribuyente	NOMBRES y APELLIDOS	FECHA VENC.	MONTO	TRIBUTOS
00000037735	JOSE MARIA RAMIREZ CARLOS	29-02-2016	7.73	IMPUESTO PREDIAL
00000037735	JOSE MARIA RAMIREZ CARLOS	29-02-2016	12.50	LIMPIEZA PUBLICA
00000037736	ETELVINA DAVILA RISCO	29-02-2016	7.59	IMPUESTO PREDIAL
00000037736	ETELVINA DAVILA RISCO	29-02-2016	12.50	LIMPIEZA PUBLICA
00000037738	MARITZA CHUQUIZUTA VILLARREAL	29-02-2016	12.50	LIMPIEZA PUBLICA
00000037742		29-02-2016	6.54	IMPUESTO PREDIAL
00000037742		29-02-2016	12.50	LIMPIEZA PUBLICA
00000037743	ANA ISABEL (POSESIONARIA) QUEPUY YAYE	29-02-2016	12.50	LIMPIEZA PUBLICA
00000037743	ANA ISABEL (POSESIONARIA) QUEPUY YAYE	29-02-2016	6.85	IMPUESTO PREDIAL

Captura de pantalla 29: Búsqueda de deudas por mes y año

El listado de los contribuyentes que se muestra anteriormente, es básicamente un método que retorna todos los registros que recibe dos parámetros: mes y año que anteriormente deberá estar autenticado para poder acceder a este recurso. Vamos a ver una manera más clara como se forma esta petición:

```

50 public function contribuyentes_por_cobrar_filtro_post()
51 {
52     $header['Authorization'] = $this->input->post('authorization');
53     if (REST_Controller::tokenIfExists($header)) {
54         try {
55             $token = $this->jwt_token_dis($header['Authorization']);
56             if ($token != false) {
57                 // verifica si el token es valido y lo desencripta (lo trae como arreglo)
58                 $output_data = $this->jwt_decode($token);
59                 $mes = $this->input->post('mes');
60                 $anio = $this->input->post('anio');
61
62                 if (!empty($mes) and !empty($anio)) {
63                     $data = $this->contribuyente_model->porCobrar($mes, $anio);
64                     if (!empty($data)) {
65                         $output_data['status'] = 'success';
66                         $output_data['message'] = $data;
67                         $this->response($output_data, REST_Controller::HTTP_OK);
68                     } else {
69                         $output_data['status'] = 'error';
70                         $output_data['message'] = 'datos no encontrados en el servidor';
71                         $this->response($output_data, REST_Controller::HTTP_UNAUTHORIZED);
72                     }
73                 } else {
74                     $output_data['status'] = 'error';
75                     $output_data['message'] = 'mes o año no vacios';
76                     $this->response($output_data, REST_Controller::HTTP_UNAUTHORIZED);
77                 }
78             } else {
79                 $output_data[$this->config->item('rest_status_field_name')] = 'portador_invalido';
80                 $output_data[$this->config->item('rest_message_field_name')] = 'portador incorrecto';
81                 $this->response($output_data, REST_Controller::HTTP_NOT_ACCEPTABLE);
82             }
83         } catch (Exception $e) {
84             $output_data[$this->config->item('rest_status_field_name')] = 'token_invalido';
85             $output_data[$this->config->item('rest_message_field_name')] = 'usted esta tratando de usar un token no
86             valido';
87             $this->response($output_data, REST_Controller::HTTP_UNAUTHORIZED);
88         }
89     }
90 }
91

```

Luego de haber dado clic en el botón Enviar Deuda se realiza el envío de la lista generada con los contribuyentes filtrada por el mes y año de deuda al banco.

Captura de pantalla 30: Envío de las deudas de los contribuyentes al banco

### 5.3.9.2. ENVIAR COBROS GLOBAL

Nos permite enviar todas las deudas al banco registradas en la base de datos de la Municipalidad de José Leonardo Ortiz.

COD. Contribuyente	NOMBRES y APELLIDOS	FECHA VENC.	Monto	Tributo	Enviar
00000037735	JOSE MARIA RAMIREZ CARLOS	28-02-2010	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	31-05-2010	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	31-08-2010	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	30-11-2010	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	28-02-2011	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	31-05-2011	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	31-08-2011	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	30-11-2011	5.40	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	29-02-2012	5.47	IMPUESTO PREDIAL	Enviar Deuda
00000037735	JOSE MARIA RAMIREZ CARLOS	31-05-2012	5.47	IMPUESTO PREDIAL	Enviar Deuda

Captura de pantalla 31: Listado de deudas de los contribuyentes



### 5.3.9.3. SUBIR ARCHIVO

Pedimos al usuario que suba un archivo .txt que retorna del banco, al subirlo modifica los datos de todas las cobranzas que se realizaron en el banco por contribuyentes que pagaron. Veamos en código esta petición:

```

114 public function enviar_archivo_x_filtro()
115 {
116     $response=array();
117     if (Session::existe_key('tbl_filtrada')) {
118         $ar=$this->create_file('CDPG.txt');
119         $empresa = "MUNICIPALIDAD DISTRITAL JOSE LEONARDO ORTIZ";
120         $longEmpresa = strlen($empresa);
121         if ($longEmpresa <= 40) {
122             for ($i=0; $i < (40 - $longEmpresa); $i++) {
123                 $empresa.=" ";
124             }
125         } else {
126             $empresa = substr($empresa, 0,40);
127         }
128         ini_set('date.timezone', 'America/Lima');
129         $fecha = date('Y-m-d',time());
130         $fechaSistema = substr($fecha, 0,4).substr($fecha, 5,2).substr($fecha, 8,2);
131
132         $cantidadRegistros = count(Session::get('tbl_filtrada'));
133         $longitudReg = strlen($cantidadRegistros);
134         $registro = "";
135         if ($longitudReg <= 9) {
136             $registro = str_pad($cantidadRegistros, 9, "0", STR_PAD_LEFT);
137         }
138
139         $acumulado = 0.0;
140         for ($i = 0; $i < count(Session::get('tbl_filtrada')); $i++) {
141             $acumulado += number_format(Session::get('tbl_filtrada')[$i]['Insol'], 2, '.', '');
142         }
143         $auxAcumulado = str_replace('.', '', number_format($acumulado, 2, '.', ''));
144         $longitudAuxAcumulado = strlen($auxAcumulado);
145         $total = "";
146         if ($longitudAuxAcumulado <= 15) {
147             $total = str_pad($auxAcumulado, 15, "0", STR_PAD_LEFT);
148         }
149
150         $libre="";
151         for ($i=0; $i < 114 ; $i++) {
152             $libre.=" ";
153         }
154
155         fwrite($ar, "CC"); //Cabecera
156         fwrite($ar, "310"); //Código de la sucursal
157         fwrite($ar, "0"); //Código de la moneda
158         fwrite($ar, "9385110"); //Número de la cuenta de la empresa afiliada (7 caracteres)
159         fwrite($ar, "C"); //Tipo de validación
160         fwrite($ar, $empresa); //Nombre de la empresa afiliada (40 caracteres)
161         fwrite($ar, $fechaSistema); //Fecha de transmisión
162         fwrite($ar, $registro); //Cantidad total de registros enviados (9 caracteres)
163         fwrite($ar, $total); //Monto total enviado (15 caracteres)
164         fwrite($ar, $libre); //Libre (144 caracteres)
165         fwrite($ar, "<br>");
166
167         /* Detalle */
168
169         $campo="";
170         for ($i=0; $i < 30 ; $i++) {
171             $campo.=" ";
172         }
173
174         $mora="";
175         for ($i=0; $i < 15 ; $i++) {
176             $mora="0";
177         }
178
179         $libre0="";
180         for ($i=0; $i < 57 ; $i++) {
181             $libre0.=" ";
182         }
183         $flag = false;
184         $s = "";
185         for ($i = 0; $i < count(Session::get('tbl_filtrada')); $i++) {
186             // fwrite($ar, "\n");
187             fwrite($ar, "00"); //Detalle
188             fwrite($ar, "310"); //Código de la sucursal
189             fwrite($ar, "0"); //Código de la moneda
190             fwrite($ar, "9385110"); //Número de la cuenta de la empresa afiliada (7 caracteres)
191
192             $codCont = Session::get('tbl_filtrada')[$i]['CodCont'];
193             $longitudCod = strlen($codCont);
194             $codContr = "";
195             if ($longitudCod <= 14) {
196                 $codContr = str_pad($codCont, 14, "0", STR_PAD_LEFT);

```

```

194 $codContr = "";
195 if ($longitudCod ≤ 14) {
196     $codContr = str_pad($codContr, 14, "0", STR_PAD_LEFT);
197     fwrite($ar, $codContr); //Código de identificación del depositante (14 caracteres)
198 }
199
200 $nombre = Session::get('tbl_filtrada')[41]['ApellidoPaterno']. " ". Session::get('tbl_filtrada')[41][
    'ApellidoMaterno']. " ". Session::get('tbl_filtrada')[41]['Nombres']. Session::get('tbl_filtrada')[41][
    'RazonSocial'];
201 $longNom = strlen($nombre);
202 if ($longNom ≤ 40) {
203     $nombre = str_pad($nombre, 40, " ");
204     fwrite($ar, $nombre); //Nombre del depositante (40 caracteres)
205 } else {
206     $nombre = substr($nombre, 0, 40);
207     fwrite($ar, $nombre); //Nombre del depositante (40 caracteres)
208 }
209
210 fwrite($ar, $campo); //Campo de información de retorno (30 caracteres)
211 fwrite($ar, "20180202"); //Fecha de emisión del cupón (8 caracteres)
212
213 $fechaDetalle = substr(Session::get('tbl_filtrada')[41]['FecVenc'], 0, 4).substr(Session::get('tbl_filtrada')
    [41]['FecVenc'], 5, 2).substr(Session::get('tbl_filtrada')[41]['FecVenc'], 8, 2);
214 fwrite($ar, $fechaDetalle); //Fecha de vencimiento del cupón (8 caracteres)
215
216 $montoTotalCont = str_replace('.', '', number_format(Session::get('tbl_filtrada')[41]['Insol'], 2, '.', ''))
    ;
217 $longMontoTotalCont = strlen($montoTotalCont);
218 $totalCont = "";
219 if ($longMontoTotalCont ≤ 15) {
220     $totalCont = str_pad($montoTotalCont, 15, "0", STR_PAD_LEFT);
221     fwrite($ar, $totalCont); //Monto del cupón (15 caracteres)
222 }
223
224 fwrite($ar, $mora); //Monto de mora (15 caracteres)
225 fwrite($ar, $libreD); //Libre (15 caracteres)
226 fwrite($ar, "<br>");
227
228 $datos=array(
229     'authorization' => 'Port '. Session::get('app_id_cjbm'),
230     'CodDeuda' => Session::get('tbl_filtrada')[41]['CodDeuda'],
231     'CodCont' => Session::get('tbl_filtrada')[41]['CodCont'],
232     'FecVenc' => Session::get('tbl_filtrada')[41]['FecVenc'],
233     'Insol' => Session::get('tbl_filtrada')[41]['Insol']
234 );

```

```

227
228 $datos=array(
229     'authorization' => 'Port '. Session::get('app_id_cjbm'),
230     'CodDeuda' => Session::get('tbl_filtrada')[41]['CodDeuda'],
231     'CodCont' => Session::get('tbl_filtrada')[41]['CodCont'],
232     'FecVenc' => Session::get('tbl_filtrada')[41]['FecVenc'],
233     'Insol' => Session::get('tbl_filtrada')[41]['Insol']
234 );
235 $actualizar_deuda = json_decode($this->cobrosModel->actualizar_deuda_a_enviado('enviar_cobros', $datos),
    true);
236 if ($actualizar_deuda['status'] == 'OK') {
237     $flag = true;
238 }
239
240 if ($flag) {
241     Session::destroy_index('tbl_filtrada');
242     fclose($ar);
243     $response['response']='Se enviaron correctamente los contribuyentes';
244 }else{
245     $response['response']='Ocurrió un error al enviar los datos';
246 }
247 }else{
248     $response['response'] = 'no hay datos que enviar';
249 }
250 echo json_encode($response);
251 }

```

En el sistema web Tesis\_Banco accederemos con credenciales a través de un formulario

**BancoTesis**

Inicia sesión para comenzar tu sesión

Usuario

Contraseña

**INGRESAR**

Copyright - 2019

Desarrollado por: Cristian Bravo y Edson Rios

Captura de pantalla 32: Inicio de sesión BancoTesis

Después de haber ingresado las credenciales correctas, el sistema nos mostrará un panel de control donde se muestra los contribuyentes que tienen que pagar.

AdminLTE	Administrador
PANEL ADMINISTRATIVO	
Principal	
Gestión de Pago	
Pagar	
Enviar Archivo	
Finalizar	
Cerrar Sesión	

Contribuyente <small>Faltan por pagar</small>					
Mostrar	10	registros por pagina	Buscar:		
CODIGO	NOMBRE	FECHA VENC.	MONTO	TRIBUTO	PAGAR
00000037742	SOC. CONYUG. AGREDA COLUNCHE JOSE EMITERIO Y GONZALES CORONADO BLANCA FANI	2017-11-30	9.94	IMPUESTO PREDIAL	PAGAR
00000037741	ELI CLAUDIO AGUILAR BURGA	2017-11-30	187.91	IMPUESTO PREDIAL	PAGAR
00000037753	WILMER GARCIA CERCADO	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037738	MARITZA CHUQUIZUTA VILLARREAL	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037755	ROSA TORRES FERNANDEZ	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037756	JUANA ROSA E HIJOS RIOJA CAMPOS	2017-11-30	29.07	IMPUESTO PREDIAL	PAGAR
00000037752	CUELLO NAYRA JESUS - GUEVARA TORRES JAIME	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037736	ETELVINA DAVILA RISCO	2017-11-30	7.92	IMPUESTO PREDIAL	PAGAR
00000037757	SOC. CONYUG. VASQUEZ FERNANDEZ JOSE CRUZ - TARRILLO VASQUEZ ROSA	2017-11-30	25.93	IMPUESTO PREDIAL	PAGAR
00000037749	ROBERTO FERNANDEZ ANYAYPOMA	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR

Mostrando pagina 1 de 157

Anterior 1 2 3 4 5 ... 157 Siguiente

Captura de pantalla 33: Listado de los contribuyentes deudores BancoTesis

El listado de los contribuyentes por pagar que se muestra anteriormente, es básicamente un método que retorna todos los registros que cumplen la condición de contribuyentes por pagar, deberá estar autenticado para poder acceder a este recurso. Vamos a ver una manera más clara como se forma esta petición

```

16 public function mostrar_clientes_deudores_post(){
17     $header['Authorization'] = $this->input->post('authorization');
18     if (REST_Controller::tokenIfExists($header)) {
19         try {
20             $token = $this->jwt_token_dis($header['Authorization']);
21             if ($token != false) {
22                 // verifica si el token es valido y lo descripta (lo trae como arreglo)
23                 $output_data = $this->jwt_decode($token);
24                 $output_data['status'] = 'success';
25                 $output_data['data'] = $this->contribuyentes_model->mostrarClientesDeudores();
26                 $this->response($output_data, REST_Controller::HTTP_OK);
27             }
28             else{
29                 $output_data[$this->config->item('rest_status_field_name')] = 'portador_invalido';
30                 $output_data[$this->config->item('rest_message_field_name')] = 'portador incorrecto';
31                 $this->response($output_data, REST_Controller::HTTP_NOT_ACCEPTABLE);
32             }
33         } catch (Exception $e) {
34             $output_data[$this->config->item('rest_status_field_name')] = 'token_invalido';
35             $output_data[$this->config->item('rest_message_field_name')] = 'usted esta tratando de usar un token no valido';
36             $this->response($output_data, REST_Controller::HTTP_UNAUTHORIZED);
37         }
38     }
39     else{
40         $output_data[$this->config->item('rest_status_field_name')] = 'acceso_prohibido';
41         $output_data[$this->config->item('rest_message_field_name')] = 'autorización no existe';
42         $this->response($output_data, REST_Controller::HTTP_FORBIDDEN);
43     }
44 }
45 }
46

```

Para realizar el pago de un contribuyente se da clic en el botón PAGAR, después nos mostrará un modal de: ¿Confirmar pago? Al dar clic en CONTINUAR se ejecuta el pago correspondiente.

AdminLTE

Principal

Gestión de Pago

Pagar

Enviar Archivo

Finalizar

Cerrar Sesión

Administrador

Inicio

Pagos

Contribuyente

Faltan por pagar

Mostrar 10 registros por página

Buscar:

CODIGO	NOMBRE	FECHA VENC.	MONTO	TRIBUTOS	PAGAR
00000037742	SOC. CONYUG. AGREDA COLUNCHE JOSE EMITERIO Y GONZALES CORONADO BLANCA FANI	2017-11-30	9.94	IMPUESTO PREDIAL	PAGAR
00000037741	ELI CLAUDIA	2017-11-30	187.91	IMPUESTO PREDIAL	PAGAR
00000037753	WILMER GARCIA	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037738	MARITZA C	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037755	ROSA TORRES	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037756	JUANA ROSA	2017-11-30	29.07	IMPUESTO PREDIAL	PAGAR
00000037752	CUELLO NAYRA JESUS - GUEVARA TORRES JAIME	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037736	ETELVINA DAVILA RISCO	2017-11-30	7.92	IMPUESTO PREDIAL	PAGAR
00000037757	SOC. CONYUG. VASQUEZ FERNANDEZ JOSE CRUZ - TARRILLO VASQUEZ ROSA	2017-11-30	25.93	IMPUESTO PREDIAL	PAGAR
00000037749	ROBERTO FERNANDEZ ANYAYPOMA	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR

Mostrando pagina 1 de 157

Anterior

1

2

3

4

5

...

157

Siguiente

Captura de pantalla 34: Realizar el pago de un contribuyente

Vamos a ver una manera más clara de cómo se forma esta petición:

```
47 public function pagar_deuda_x_banco_post(){
48     $header['Authorization'] = $this->input->post('authorization');
49     if (REST_Controller::tokenIfExists($header)) {
50         try {
51             $token = $this->jwt_token_dis($header['Authorization']);
52             if ($token != false) {
53                 // verifica si el token es valido y lo desencripta (lo trae como arreglo)
54
55                 $codigo = isset($_POST['codDeuda']) ? $_POST['codDeuda'] : null;
56                 $fecha = isset($_POST['fecha']) ? $_POST['fecha'] : null;
57                 $hora = isset($_POST['hora']) ? $_POST['hora'] : null;
58
59                 $this->contribuyentes_model->changeStatusDeuda($codigo, $fecha);
60                 $this->contribuyentes_model->changeStatusCobranza($codigo, $fecha, $hora);
61                 //validaremos si se han logrado pagar los contribuyentes a traves del codigoDeuda y su estado
62                 $dataDeuda = $this->contribuyentes_model->checkPaymentDeudaByBanco($codigo);
63                 $dataCobranza = $this->contribuyentes_model->checkPaymentCobranzaByBanco($codigo);
64                 $output_data = $this->jwt_decode($token);
65
66
67                 if ($dataDeuda['CodDeuda'] == $codigo && $dataCobranza['CodDeuda'] == $codigo) {
68                     $output_data['data'] = array(
69                         $dataDeuda['Nombres'],
70                         $dataDeuda['ApellidoPaterno'],
71                         $dataDeuda['Insol'],
72                         $dataCobranza['CodDeuda'],
73                         $dataDeuda['NombTrib']
74                     );
75                     $this->response($output_data, REST_Controller::HTTP_OK);
76                 } else {
77                     $output_data['data'] = array("transacción incompleta");
78                     echo json_encode(array("transacción incompleta"));
79                     $this->response($output_data, REST_Controller::HTTP_OK);
80                 }
81                 $this->response($output_data, REST_Controller::HTTP_OK);
82
83             } else {
84                 $output_data[$this->config->item('rest_status_field_name')] = 'portador_invalido';
85                 $output_data[$this->config->item('rest_message_field_name')] = 'portador incorrecto';
86                 $this->response($output_data, REST_Controller::HTTP_NOT_ACCEPTABLE);
87             }
88         } catch (Exception $e) {
89
90     }
```

Después de haber confirmado el pago se mostrará un mensaje “Pago exitoso del contribuyente”.

AdminLTE

Principal

Gestión de Pago

Pagar

Enviar Archivo

Finalizar

Cerrar Sesión

Administrador

Contribuyente

Faltan por pagar

Inicio

Pagos

Pago Exitoso del contribuyente con codigo de deuda 2478330 - Tributo: IMPUESTO PREDIAL

Mostrar 10 registros por pagina

Buscar:

CODIGO	NOMBRE	FECHA VENC.	MONTO	TRIBUTO	PAGAR
00000037741	ELI CLAUDIO AGUILAR BURGA	2017-11-30	187.91	IMPUESTO PREDIAL	PAGAR
00000037753	WILMER GARCIA CERCADO	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037738	MARITZA CHUQUIZUTA VILLARREAL	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037755	ROSA TORRES FERNANDEZ	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037756	JUANA ROSA E HIJOS RIOJA CAMPOS	2017-11-30	29.07	IMPUESTO PREDIAL	PAGAR
00000037752	CUELLO NAYRA JESUS - GUEVARA TORRES JAIME	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037736	ETELVINA DAVILA RISCO	2017-11-30	7.92	IMPUESTO PREDIAL	PAGAR
00000037757	SOC. CONYUG. VASQUEZ FERNANDEZ JOSE CRUZ - TARRILLO VASQUEZ ROSA	2017-11-30	25.93	IMPUESTO PREDIAL	PAGAR
00000037749	ROBERTO FERNANDEZ ANYAYPOMA	2017-11-30	6.08	IMPUESTO PREDIAL	PAGAR
00000037750	FLOR YULIANA VASQUEZ CHIRINOS	2017-11-30	15.29	IMPUESTO PREDIAL	PAGAR

Mostrando pagina 1 de 157

Anterior

1

2

3

4

5

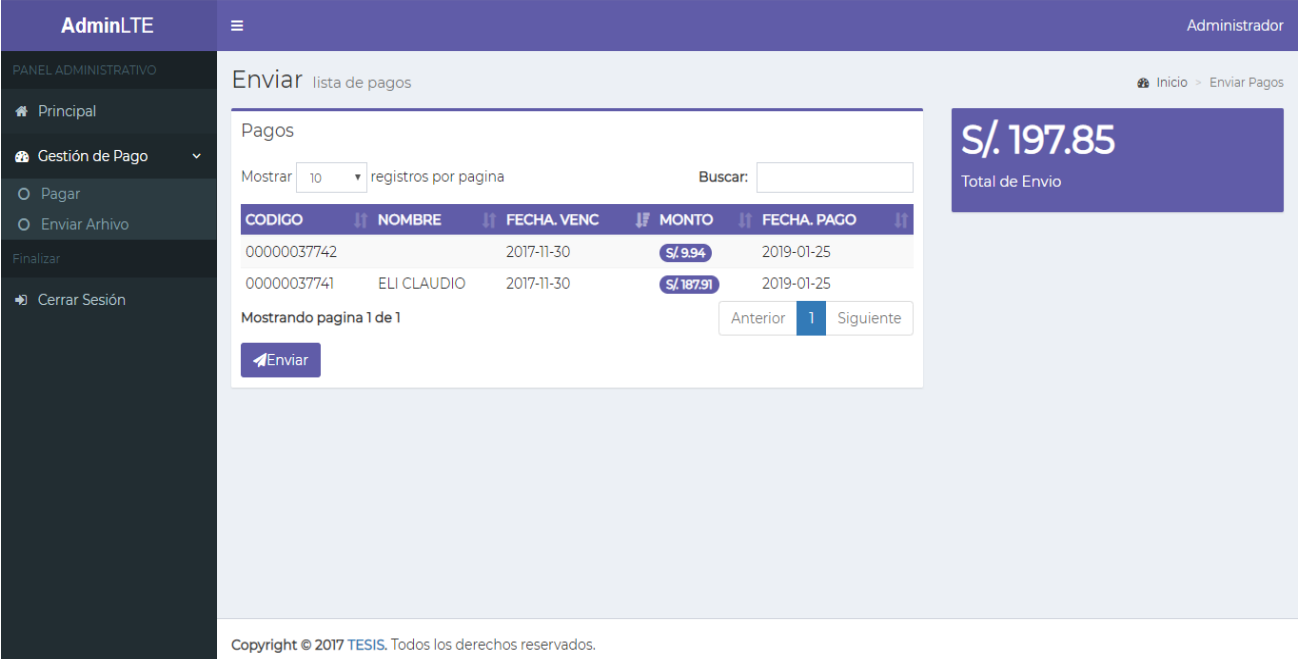
...

157

Siguiente

Captura de pantalla 35: Pago realizado de un impuesto del contribuyente

En esta segunda parte se muestra el listado de pagos, donde se muestra el código, monto y la fecha de pago que posteriormente se enviará a la Municipalidad de José Leonardo Ortiz en formato .txt.



AdminLTE

Administrador

PANEL ADMINISTRATIVO

- Principal
- Gestión de Pago
  - Pagar
  - Enviar Archivo
- Finalizar
- Cerrar Sesión

Enviar lista de pagos

Inicio > Enviar Pagos

Pagos

Mostrar 10 registros por pagina

Buscar:

CODIGO	NOMBRE	FECHA. VENC	MONTO	FECHA. PAGO
00000037742		2017-11-30	S/. 99.94	2019-01-25
00000037741	ELI CLAUDIO	2017-11-30	S/. 187.91	2019-01-25

Mostrando pagina 1 de 1

Anterior 1 Siguiente

Enviar

S/. 197.85

Total de Envio

Copyright © 2017 TESIS. Todos los derechos reservados.

Captura de pantalla 36: Listado de pagos para enviar a la Municipalidad de José Leonardo Ortiz

Vamos a ver una manera más clara como se forma esta petición:

```

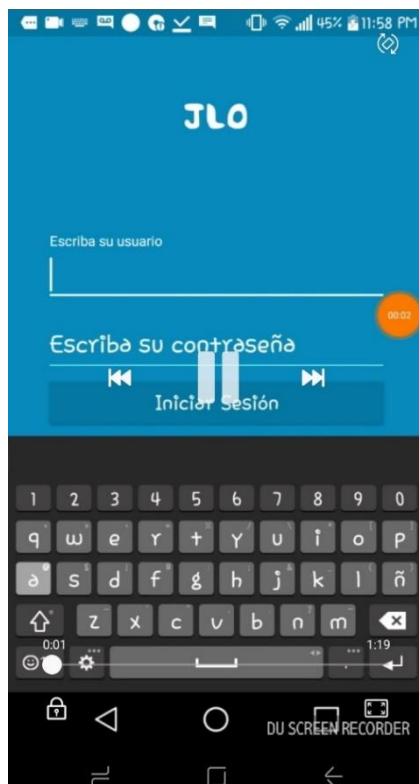
103 public function mostrar_clientes_no_deudores_post(){
104     $header['Authorization'] = $this->input->post('authorization');
105     if (REST_Controller::tokenIfExists($header)) {
106         try {
107             $token = $this->jwt_token_dis($header['Authorization']);
108             if ($token != false) {
109                 // verifica si el token es valido y lo descripta (lo trae como arreglo)
110                 $output_data = $this->jwt_decode($token);
111                 $output_data['status'] = 'success';
112                 $output_data['data'] = $this->contribuyentes_model->mostrarClientesNoDeudores();
113                 $this->response($output_data, REST_Controller::HTTP_OK);
114             }
115             else{
116                 $output_data[$this->config->item('rest_status_field_name')] = 'portador_invalido';
117                 $output_data[$this->config->item('rest_message_field_name')] = 'portador incorrecto';
118                 $this->response($output_data, REST_Controller::HTTP_NOT_ACCEPTABLE);
119             }
120         } catch (Exception $e) {
121             $output_data[$this->config->item('rest_status_field_name')] = 'token_invalido';
122             $output_data[$this->config->item('rest_message_field_name')] = 'usted esta tratando de usar un token no
123                 valido';
124             $this->response($output_data, REST_Controller::HTTP_UNAUTHORIZED);
125         }
126     }
127     else{
128         $output_data[$this->config->item('rest_status_field_name')] = 'acceso prohibido';
129         $output_data[$this->config->item('rest_message_field_name')] = 'autorización no existe';
130         $this->response($output_data, REST_Controller::HTTP_FORBIDDEN);
131     }
132 }
133

```

Luego de haber dado clic en el botón Enviar se realiza el envío de pagos a la Municipalidad de José Leonardo Ortiz. Vamos a ver una manera más clara como se forma esta petición:

```
168 public function enviar_pago_desde_banco_post()
169 {
170     $header['Authorization'] = $this->input->post('authorization');
171     if (REST_Controller::tokenIfExists($header)) {
172         try {
173             $token = $this->jwt_token_dis($header['Authorization']);
174             if ($token != false) {
175                 // verifica si el token es valido y lo desencripta (lo trae como arreglo)
176                 $output_data = $this->jwt_decode($token);
177                 $output_data['status'] = 'success';
178                 $codDeuda = isset($_POST['codDeuda'])? $_POST['codDeuda']: null;
179                 $this->contribuyentes_model->changeStatusDeudaSend($codDeuda);
180                 $this->contribuyentes_model->changeStatusCobranzaSend($codDeuda);
181                 $this->response($output_data, REST_Controller::HTTP_OK);
182             }
183             else{
184                 $output_data[$this->config->item('rest_status_field_name')] = 'portador_invalido';
185                 $output_data[$this->config->item('rest_message_field_name')] = 'portador incorrecto';
186                 $this->response($output_data, REST_Controller::HTTP_NOT_ACCEPTABLE);
187             }
188         } catch (Exception $e) {
189             $output_data[$this->config->item('rest_status_field_name')] = 'token_invalido';
190             $output_data[$this->config->item('rest_message_field_name')] = 'usted esta tratando de usar un token no
191             valido';
192             $this->response($output_data, REST_Controller::HTTP_UNAUTHORIZED);
193         }
194     }
195     else{
196         $output_data[$this->config->item('rest_status_field_name')] = 'acceso_prohibido';
197         $output_data[$this->config->item('rest_message_field_name')] = 'autorización no existe';
198         $this->response($output_data, REST_Controller::HTTP_FORBIDDEN);
199     }
200 }
```

### 5.3.10. APLICACIÓN MÓVIL

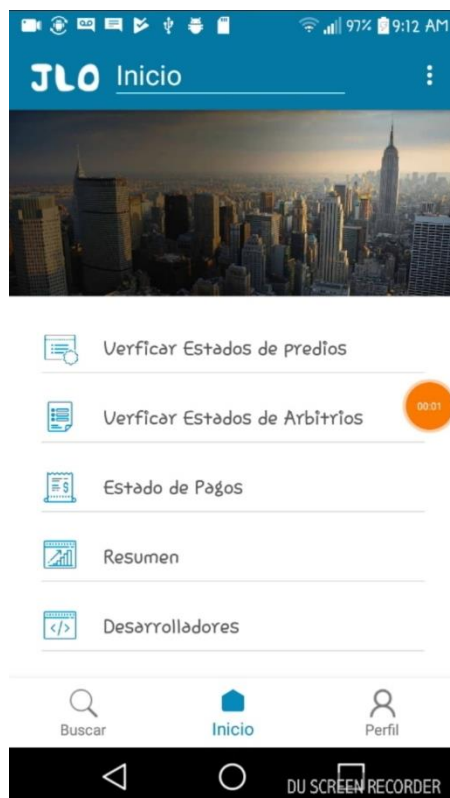


Captura de pantalla 37: Inicio de Sesión



Captura de pantalla 38: Perfil de Usuario

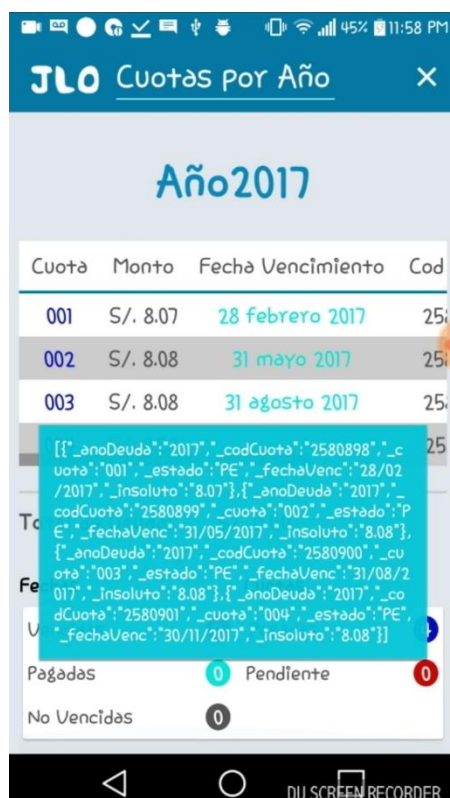




Captura de pantalla 39: Menú de Opciones



Captura de pantalla 40: Vista de Estado de Predio



Captura de pantalla 41: Cuotas por Año



Captura de pantalla 42: Resumen por Año





Captura de pantalla 43: Vista de Estado Arbitrios



Captura de pantalla 44: Vista de Pagos



Captura de pantalla 45: Web de la Municipalidad de José Leonardo Ortiz

**CAPÍTULO VI**  
**RESULTADO, COSTOS**  
**Y BENEFICIOS**

## VI. ANÁLISIS DE COSTOS

### 6.1.COSTO DE PERSONAL

ESPECIALISTAS	PRECIO	TIEMPO (MES)	TOTAL
Analista Programador	S/. 3,500.00	3	S/. 10,500.00
Diseñador	S/. 3,000.00	3	S/. 9,000.00
<b>T O T A L</b>			<b>S/. 19,500.00</b>

Tabla 13: Costo de Personal  
Autor: Elaboración propia

### 6.2.COSTO DE SERVICIOS

SERVICIOS	PRECIO
Hosting por mes	S/. 20.00
Internet por mes	S/. 200.00
Cloud (nube Azure) por mes	S/. 680.00
<b>T O T A L</b>	<b>S/. 900.00</b>

Tabla 14: Costo de Servicios  
Autor: Elaboración propia

### 6.3. COSTO DE HARDWARE

HARDWARE	CANTIDAD	PRECIO
PC de escritorio	3	S/. 2,400.00
Servidores	0	S/.0.00
<b>T O T A L</b>		<b>S/. 7,200.00</b>

Tabla 15: Costo de Hardware  
Autor: Elaboración propia

Los servidores se encuentran abastecidos por la entidad de la Municipalidad de José Leonardo Ortiz; por lo tanto, no involucra un costo.

### 6.4.COSTO DE SOFTWARE

SOFTWARE	PRECIO
SQL Server	S/. 4,600.00
<b>T O T A L</b>	<b>S/. 4,600.00</b>

Tabla 16: Costo de Software  
Autor: Elaboración propia

## 6.5.RESUMEN DE COSTOS DE INVERSIÓN

RESUMEN DE COSTOS DE INVERSIÓN	
Descripción	Sub Total
Costo de personal	S/.19,500.00
Costo de servicios	S/.900.00
Costo de hardware	S/.7,200.00
Costo de software	S/. 4,600.00
<b>T O T A L</b>	<b>S/. 32,200.00</b>

Tabla 17: Resumen de costos de inversión

Autor: Elaboración propia

## 6.6.VIABILIDAD DEL PROYECTO

INVERSIÓN S/. 32,000.00

TASA DE DESCUENTO: 10%

PRODUCTO	INVERSIÓN (S/. )	1° MES (S/. )	2° MES (S/. )	3° MES (S/. )	VAN (S/. )	TIR
APLICATIVO MÓVIL	-32,000.00	14,000.00	15,000.00	16,000.00	5,145.00	19 %

✚ En el primer mes se tuvo un monto de S/. 14,000.00 por:

	Mes	Precio
Personal	1	S/. 2,850.00
Alquiler de impresora	1	S/. 7,000.00
Alquiler de laptop	1	S/. 3,800.00
Papel bond	1	S/. 350.00
<b>T O T A L</b>		<b>S/. 14,000.00</b>

✚ En el segundo mes se tuvo un monto de S/. 15,000.00 por:

	Mes	Precio
Personal	1	S/. 2,850.00
Alquiler de impresora	1	S/. 7,000.00
Alquiler de laptop	1	S/. 3,800.00
Papel bond	1	S/. 1,350.00
<b>T O T A L</b>		<b>S/. 15,000.00</b>

✚ En el tercer mes se tuvo un monto de S/. 16,000.00 por:

	Mes	Precio
Personal	1	S/. 2,850.00
Alquiler de impresora	1	S/. 7,000.00
Alquiler de laptop	1	S/. 3,800.00
Papel bond	1	S/. 2,350.00
<b>T O T A L</b>		<b>S/. 16,000.00</b>

El proyecto es viable porque el VAN nos arroja un valor positivo y la TIR es superior a la tasa de descuento.

**CAPÍTULO VII**  
**CONCLUSIONES Y**  
**RECOMENDACIONES**

## VII. CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

Con el desarrollo de la web service y el aplicativo móvil, permitió facilitar el acceso a la información del área de rentas a todos los ciudadanos Leonardinos.

Utilizamos Scrum, como metodología ágil, que nos facilitó a realizar entregas a corto plazo según los requisitos más prioritarios en ese momento, teniendo una mejor flexibilidad y adaptación respecto a las necesidades del cliente o cambios que puedan suceder durante el desarrollo del proyecto.

Se diseñaron interfaces atractivas, flexibles y fáciles de utilizar para que el usuario pueda comunicarse con la aplicación de forma sencilla e intuitiva posible.

En cuanto a las herramientas de software: Los Tokens son una magnífica forma de administrar la autenticación de los usuarios, pues nos permitió crear de forma fácil un mecanismo para comunicarse de forma segura entre el cliente y el servidor, además, de que también nos permitió mantener autenticado a un usuario por un tiempo determinado, evitando tener que iniciar sesión cada vez que entre a nuestra aplicación.

En cuanto a la seguridad, la librería JWT nos ha proporcionado la facilidad de hacer toda la manipulación de todo el proceso del inicio de sesión y de la restricción a los APIs de nuestro servicio web, enviando cualquier dato del cliente dentro del Token para que el servidor pueda obtener mucha más información de nosotros.

Finalmente, en cuanto a la estructura de costos que implicará la realización de la propuesta planteada; el proyecto, según lo analizado, es viable debido a que el VAN nos da como resultado un valor positivo y la TIR es superior a la tasa de descuento.

### RECOMENDACIONES

Realizar pruebas de Stress para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos.

Las pruebas enfocadas de determinar o validar las características de desempeño de un sistema expuesto a condiciones más allá de las anticipadas para producción. Enfatizando aspectos como la robustez, disponibilidad, manejo de errores, etc., bajo una carga pesada, el objetivo de estas pruebas es de asegurar que el sistema operativo no colapse cuando disponga de pocos recursos o exista una gran concurrencia de usuarios.



## REFERENCIAS BIBLIOGRÁFICAS

- Academia Android. (2015). *AcademiaAndroid.com*. Obtenido de <https://academiaandroid.com/android-studio-v1-caracteristicas-comparativa-eclipse/>
- Aguirre Rodríguez, C. E. (2012). *Implementación de un web services para mejorar la calidad del servicio aplicado a un sistema ERP del grupo COOPSOL - Lima*. Trujillo: Universidad Nacional de Trujillo.
- Albaladejo, X. (2011). Que es Scrum. En *Proyectos Ágiles*.
- Anónimo. (2011). *Blogs*. Obtenido de <https://mantenimientosdeunapc.blogspot.com/2011/11/que-es-xampp-y-para-que-sirve.html>
- Azaustre, C. (2015). *carlosazaustre.es*. Obtenido de ¿Qué es la autenticación basado en Token?: <https://carlosazaustre.es/que-es-la-autenticacion-con-token/>
- Brito Moreno, J. E. (2011). *Implementar el consumo de servicios vía web services de la información almacenada en una base de datos de una biblioteca digital perteneciente a una institución educativa de nivel medio*. Ecuador: Universidad de Guayaquil.
- Cardoso de Miranda, E. A., & Muñoz Cañavate, A. (2015). *Los sitios web como servicios de información al ciudadano: Un estudio sobre los 308 ayuntamientos de Portugal*. Lisboa – Portugal: Universidade Lusófona de Humanidades e Tecnologias.
- Carrión, R., Noriega, A., & Del Castillo, D. (2019). *Usando XAMPP con Bootstrap y WordPress*. Rama Solutions.
- Díaz Acuña, A. C., & Díaz Vásquez, C. E. (2011). *Desarrollo e implementación de un sistema informático la metodología Scrum para optimizar los procesos en el área de ventas y almacén de la botica San Juan – Distrito de Pátapo Chiclayo - Lambayeque*. Lambayeque: Universidad Nacional Pedro Ruiz Gallo.
- Fontán, M. (2012). *Asociación Desarrolladores Web De España*. Obtenido de <http://www.adwe.es/codigo/codeigniter-framework-php-desarrollo-aplicaciones-web>
- Grupo PHP. (2019). *PHP.NET*. Obtenido de <http://php.net/manual/es/intro-what-is.php>
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, P. (2016). *Metodología de la Investigación 6ta Edición*. Mexico D.F.: Mc Graw Hill Education.
- James. (2018). *Hermosa programación*. Obtenido de <http://www.hermosaprogramacion.com/2015/02/android-volley-peticiones-http/>
- Javier Paredes, J. (2011). *Desarrollo de una aplicación web para la administración de la información del centro oftalmológico Dr. Martin Inoquio Oliva*. Lambayeque: Universidad Nacional Pedro Ruiz Gallo.
- Palacios Tolón, Á., & Moreno Sama, E. (2019). *La Madriguera Bit*. Obtenido de <https://lamadriguerabit.com/articulos/que-es-postman/>
- Paszniuk, R. (2016). *Programacion.com.py*. Obtenido de ¿Qué es Json Web Token (JWT)?: <https://www.programacion.com.py/varios/que-es-json-web-token-jwt>

- Rodríguez Chávez, J. C. (2016). *Sistema de información web para la mejora de la gestión de la subgerencia de abastecimiento de la Municipalidad Provincial de Utcubamba*. Trujillo: Universidad Nacional de Trujillo.
- Rouse, M. (2015). *Techtarget*. Obtenido de Techtarget: <https://searchdatacenter.techtarget.com/es/definicion/SQL-Server>
- Severino Saenz, L. J., & Requejo Delgado, V. I. (2014). *Portafolio electrónico para facilitar la información de la formación académica de los estudiantes de la Escuela Profesional de Ingeniería en Computación e Informática de la Universidad Nacional Pedro Ruiz Gallo*. Lambayeque: Universidad Nacional Pedro Ruiz Gallo.
- Tecon. (2019). Obtenido de Que es Microsoft Azure y cómo funciona: <https://www.tecon.es/que-es-microsoft-azure-como-funciona/>
- Trigas Gallegos, M. (2011). Metodología SCRUM. En *Gestion de proyectos Informaticos* (pág. 5).