

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

Departamento de Ingeniería Eléctrica y Computación



IMPLEMENTACIÓN DE UN SISTEMA QUE INTEGRA REGISTROS
DE ASISTENCIA DESDE DIFERENTES PUNTOS DE LA
REPÚBLICA Y WEB SERVICE

Reporte Técnico de Investigación presentado por:

Lino Andres Gallegos Garcia 105933

Requisito para la obtención del título de
INGENIERO EN SISTEMAS COMPUTACIONALES

Hugo Brito Holguín

Ciudad Juárez, Chihuahua

Noviembre 2019

Ciudad Juárez, Chihuahua a 30 oct. 2019

Asunto: Liberación de Asesoría

Mtro. Ismael Canales Valdiviezo
Jefe del Departamento de Ingeniería
Eléctrica y Computación
Presente. -

Por medio de la presente me permito comunicarle que después de haber realizado las asesorías correspondientes al reporte técnico **Implementación de un sistema que integra registros de asistencia desde diferentes puntos de la república y web service**, del alumno **Lino Andres Gallegos García** de la Licenciatura en Ingeniería en Sistemas Computacionales, considero que lo ha concluido satisfactoriamente, por lo que pueden continuar con los trámites de titulación intracurricular.

Sin más por el momento, reciba un cordial saludo.

Atentamente



Asesor

Hugo Brito Holguín

Ccp.
Coordinador del programa de sistemas computacionales
Lino Andres Gallegos García
Archivo



Ciudad Juárez, Chihuahua a 6 de Noviembre de 2019

Asunto: Autorización de publicación

C. Lino Andrés Gallegos García

Presente.-

En virtud de que cumple satisfactoriamente los requisitos solicitados, informo a usted que se autoriza la impresión del proyecto de Implementación de un sistema que integre registros de asistencia desde diferentes puntos de la república y web service, para presentar los resultados del proyecto de titulación con el propósito de obtener el título de Licenciado en Ingeniería en Sistemas Computacionales.

Sin otro particular, reciba un cordial saludo.



Mtra. Ivonne Haydee Robledo Portillo
Profesor titular de Seminario de Titulación II

Declaración de Originalidad

Yo Lino Andres Gallegos García declaro que el material contenido en esta publicación fue generado con la revisión de los documentos que se mencionan en la sección de referencias y que la solución desarrollada es original y no ha sido copiada de ninguna otra fuente, ni ha sido usada para obtener otro título o reconocimiento en otra Institución de Educación Superior.



Lino Andrés Gallegos Garcia

Agradecimientos

Agradezco a Dios por la experiencia de vivir a mi modo en su libertad, por reunirme con personas extraordinarias y que llevo en mi corazón. Personas de quienes he aprendido mucho: en el amor, en lo espiritual, musical, formación personal y profesional. Los amo. Agradezco a Lorenza por enseñarme a escribir debidamente. Y, Por último, a Joel Gaspar, Juan Chacón y a mi asesor Hugo Brito Holguín porque a pesar de no tener responsabilidad de acompañarme y dirigirme durante este proyecto decidieron hacerlo.

Dedicatoria

A mis papás Lino Gallegos Reyes y Rosa Isela García Cabral, a mi hermano Lino Alberto Gallegos, a mi novia Victoria Escorza y a Hunu por su amor incondicional y su paciencia al enseñarme.

A mis amigos que me compartieron sus experiencias de titulación, y que siempre tuvieron la visión de que también llegaría ahí. Por último, a la maestra Ivonne Haydee Robledo Portillo, por darme las herramientas para armar este proyecto y a guardar este último semestre como un tesoro único.

Índice de Contenido

Agradecimientos	v
Dedicatoria	vi
Índice de figuras.....	viii
Índice de tablas	viii
Resumen.....	ix
Palabras clave.....	1
Summary	1
I. Planteamiento del problema	2
1.1 Antecedentes	2
1.2 Definición del problema.....	4
1.3 Objetivo general.....	4
1.4 Objetivos específicos	4
1.5 Justificación	5
II. Marco Referencial	6
2.1 Marco Teórico.....	6
2.2 Reloj checador	6
2.1.1 ZKTeco ZKSoftware reloj checador de huella digital y control de acceso	7
2.1.2 Timework reloj checador	7
2.3 Las VPN.....	8
2.3 Introducción a las API y los Web Service	10
2.4 Herramientas de bases de datos.....	12
2.4 Servidores	13
III. Desarrollo del proyecto.....	14
3 Metodología	14
3.1 Recopilación de requisitos y especificaciones	15
3.2 Documento de análisis funcional	16
3.3 Arquitectura del sistema	17
3.4 Codificación	18
IV. Resultados.....	19
4.1 Presentación de resultados	19

4.2 Pruebas orientadas al servidor	20
4.3 Pruebas orientadas al web service.....	21
V. Conclusiones	24
5.1 Conclusiones del objetivo establecido	24
5.2 Recomendaciones a futuro	25
Referencias.....	26
Anexos	27

Índice de figuras

FIGURA 2. 1 MODELO ZK LX-50 [HTTP://WWW.IBIX.COM/ZKTECO-LX50].....	7
FIGURA 2. 2 TIMEWORK STA 4 [HTTP://WWW.RELOJCHECADOR.COM/FINGERPRINTSCANNERS.PHP].....	8
FIGURA 2. 3 FUNCIONAMIENTO VPN [HTTPS://WWW.XATAKA.COM]	9
FIGURA 2. 4 FUNCIONAMIENTO DE WEB SERVICE [HTTPS://STACKOVERFLOW.COM/QUESTIONS/226108/WHAT-IS-A-WEB-SERVICE-IN-PLAIN-ENGLISH]	10
FIGURA 2. 5 SERVIDOR Y SU FUNCIÓN [HTTPS://WWW.CODIGOHOSTING.COM/SERVIDOR-COMPARTIDO-VENTAJAS-DESVENTAJAS/].....	13
FIGURA 3. 1 MODELO V O DE CUATRO NIVELES [HTTP://INGENIERIADESFTWARE.MEX.TL/61885_MODELO-V.HTML]	14

Índice de tablas

TABLA 1 COSTOS Y COMPATIBILIDAD DE BASES DE DATOS. [ELABORACIÓN PROPIA]	12
TABLA 2 DATOS REQUERIDOS POR EL SISTEMA DE NÓMINAS [ELABORACIÓN PROPIA].....	15

Resumen

A lo largo del camino de una empresa que va creciendo surgen necesidades tecnológicas que automaticen procesos que en muchos casos parte es en computadora y otra parte a mano, como solía ser en la empresa GMEX (Grupo Maquirrental Express) que a partir de su crecimiento surge este proyecto, como primer instancia en el capítulo 1, se plantea el proyecto de implementar un sistema de nóminas completo (Nomifácil), debido a que el utilizado hasta ese momento carecía de procesos que resolvieran las necesidades de la empresa. Se investigaron algunos artículos donde empresas implementaron sistemas de nóminas de acuerdo con sus necesidades, y la facilidad con la que ahora pueden trabajar los departamentos involucrados. Lo que nos lleva a explicar en un segundo capítulo lo que formara el proyecto en su totalidad como lo son las redes vpn site to site, bases de datos, web service entre algunos otros términos teóricos y su funcionamiento. En el capítulo 3 se desglosa fase a fase la metodología utilizada y la importancia de utilizar la metodología V para esta implementación. Un cuarto capítulo nos habla de las pruebas realizadas a nuestro proyecto unas enfocadas al servidor, y otras enfocadas al *web service*, ya que uno sin el otro no alcanzaría los objetivos planteados. Por último, pero no menos importante en el capítulo 5 se fija una visión a futuro del proyecto, es decir, en el camino del inicio al término del proyecto surgen nuevas problemáticas que abren el panorama de necesidades a resolver, y se explican aquellas modificaciones necesarias para que el proyecto tenga un mayor alcance.

Palabras clave

Web service, redes vpn site to site, reloj checador, metodología V, servidor centralizado.

Summary

Along the way of the growth of companies, technological needs arise when automatic processes required computers and or handwriting, as it used to be in the GMEX company (Maquirrental Express Group). From its growth, problems and objectives need to achieve. All the information that we need to solve these issues addressed in this project. For example, in chapter 1, we can see the plan of implementing a complete payroll system (Nomifácil), this procedure is proposed because the process they used in the past lacked processes that met the needs of the company. In the investigation, articles were found and studied as examples of how other companies use payroll procedures according to their needs. This investigation leads us to chapter 2, where we explain how the project will form in its totality, such as VPN site to site networks, databases, web service, etc., besides the theoretical terms and their operation. In chapter 3, a breakdown is made phase by phase about the V methodology and its influence on this project. The test phase leads us to episode 4, where we applied two varieties of experiments, some focused on the server, and others on the web service, and the fact that one without the other does not reach the objectives set. Finally, in Chapter 5, we can see a future vision of the project, where new problems could arise in and open us the scenery of possible needs to be solved. Also, we discuss the proposal of modifications to complement and improve the project.

I. Planteamiento del problema

En este primer capítulo se detalla aquello que definió el proyecto, cuando la empresa, Grupo Maquirental Express (en adelante GMEX), analizó el sistema que utiliza para realizar la nómina, con el fin de implementar uno que automatice y mejore los tiempos en que se lleva a cabo la misma. Se plantea el análisis que se realizó por parte del departamento de sistemas y se detalla los objetivos planteados para alcanzar en tiempo y forma las herramientas, así como los datos con los que se debe de contar para que la nueva nómina, que es externa a la empresa, pueda brindarnos el servicio sin necesidad de invertir tanto.

1.1 Antecedentes

La empresa GMEX (Grupo Maquirrental Express) dedicada a la renta y venta de equipos de montacargas, lleva aproximadamente 25 años laborando en Chihuahua, y unos cuantos años menos en distintas ciudades del país, estas son: Silao, Querétaro, y Ciudad Juárez. Como en todas las compañías se necesita registrar las horas laboradas por empleado, para calcular su salario. Para esto, es de suma importancia contar con el registro de entrada y salida de manera confiable, precisa y accesible en cualquier momento. En GMEX se llevan a cabo estos registros por medio de tarjeta de proximidad, que permite en cada sucursal llevar un historial electrónico, pero exclusivamente son accesibles en cada una de las sucursales, no permite compartirlo a una sucursal matriz de manera sencilla para globalizar el total de gastos que se realiza en todas las sucursales.

Establecer la centralización de los registros mencionados permitirá que el departamento de contabilidad establecida en la sucursal matriz pueda saber con claridad que activos

estarían utilizando para pagar salarios, impuestos, incapacidades, entre otras, de todas las sucursales y se desenvuelvan mejor.

Un ejemplo de sistemas de entradas/salidas de trabajadores es el Sistema Integral Para Administración de Personal, en el que se trabajó en 2016 en la Universidad de Veracruz. El cual al igual que en la empresa GMEX se necesitaba llevar un chequeo completo de trabajadores, en el artículo comentan un aproximado de 4,983 empleados, entre cortadores de caña, obreros, empleados, vigilancia, etcétera. para ello se propuso el sistema mencionado con su registro de entrada/salida, número de seguro social, perfil, días trabajados, altas o bajas, entre otros [1] el cual en este caso logró el objetivo de realizar tareas más fácilmente en cuanto a las nóminas y ahorro de espacios físicos para la documentación.

Enfocado a los servicios web o *web services* [2], se encontró el artículo llamado Los *web services* y características de calidad. Que nos expresa precisamente de esta novedad para las empresas que es el utilizar *web services*, y cómo estos deben ser de calidad ya que frente a las demás tecnologías aportan interoperabilidad entre las aplicaciones de software, que fomentan los estándares y protocolos basados en texto, y lo más importante es que permiten que servicios y software de diferentes compañías ubicados en diferentes lugares geográficos pueden ser combinados de manera fácil para brindar servicios juntos.

1.2 Definición del problema

La empresa GMEX lleva de manera electrónica los registros de entradas y salidas más no su proceso de nóminas, ya que para circular salarios se hace de manera tradicional, es decir en papel, lo que conlleva a generar excesivamente archivo muerto. Además, es complicado poder compartir estos datos a un departamento de contabilidad central.

1.3 Objetivo general

Implementar un sistema de cómputo que permita centralizar la información de diferentes relojes checadores en una base de datos dentro de un servidor central capaz de transferir esa misma información a un sistema de nóminas externo que se desea implementar. Con la finalidad de optimizar los procesos de nómina, disminuir espacios físicos y brindar seguridad a la infraestructura informática de la empresa.

1.4 Objetivos específicos

- Centralización de información de checadas, para el acceso rápido y oportuno de información desde cualquier sucursal y así reducir tiempos/costos que esto genera.
- Brindar seguridad a la información compartida, para que se pueda trabajar de manera eficiente y sin riesgos, con la finalidad de que solo se tenga acceso a la información que nos solicita.
- Desarrollo de herramienta web para el envío de las checadas, con el objetivo de alimentar el sistema de nóminas que se va a implementar en la empresa.

1.5 Justificación

Es necesario automatizar el esfuerzo que realiza el departamento de recursos humanos y contabilidad, así como reducir espacios físicos donde almacenan esta información. Por lo que el proyecto propuesto busca que mediante el uso de las tecnologías disponibles dentro de la empresa y el uso de web service, se logre centralizar información de entrada/salida de trabajadores.

II. Marco Referencial

El proyecto está compuesto por diferentes partes que trabajan en conjunto para que todo se realice de manera correcta, el capítulo explica la parte teórica que nos ayudará a entender la participación de cada una de ellas. Conceptos como Reloj checador, red VPN site to site, bases de datos, API y web service, son necesarios de aclarar para poder definir qué compone este proyecto.

2.1 Marco Teórico

A continuación, se abordan los conceptos teóricos necesarios para la realización del proyecto y que le dan sustento al mismo. Aspectos como red vpn punto a punto, servidores, algunos sistemas de reloj checador, bases de datos y por último el api web, son los temas de interés ya que estos son los que como engranes dan forma a nuestro sistema en su totalidad.

2.2 Reloj checador

Desde hace casi un siglo, se ha necesitado un control o registro de las horas que un trabajador labora, desde un oficinista, un minero, o hasta un conductor, se tiene que probar de alguna manera si alguien asistió o no al trabajo. Como uso secundario de este registro se puede definir por ejemplo cuanto se le debe pagar o descontar. Para lo mencionado anteriormente se crearon las tecnologías del reloj checador.

Estos han ido cambiando para cumplir con mayor eficacia su trabajo, en sus inicios funcionaban agujerando tarjetas e imprimiendo la hora exacta en la que el trabajador comenzaba la labor, viejos tiempos que han cambiado para bien, pues ahora solo con introducir la huella digital, pasar una tarjeta de proximidad o acercar el rostro se registra en una base de datos mucha más que la información mencionada anteriormente (hora y fecha). Este registro de cada trabajador se define como “checada”.

La tarea del reloj checador en su sistema no ha sido sencilla, ya que los dispositivos actuales no solo se limitan imprimir un dato y olvidarlo, sino que recaban, califican y analizan la información recibida para llevar un control de asistencia de los empleados y de este modo facilitar acciones de ciertos departamentos, implementando

acciones que apoyen a la productividad de los empleados, todo a partir de un empleado que acerca su índice, su rostro o tarjeta de proximidad, según sea la tecnología del reloj.

2.1.1 ZKTeco ZKSoftware reloj checador de huella digital y control de acceso

ZKTeco es la empresa líder en México de relojes checadores con tecnología biométrica, es decir, reconocimiento facial y huella digital. Brindan su servicio a oficinas, sucursales o PYMES (pequeñas y medianas empresas) brindan respaldos, servicio, soporte y software personalizado como lo es el sistema IBIX que es un sistema de catálogos, sincronización, recolector de información, seguridad, reportes, clasificaciones de trabajadores por empresa. Entre algunas de las funcionalidades existe incluso la personalización e impresión de gafetes. Todo en conjunto, algunas de las especificaciones y el diagrama de aplicación lo podemos ver en la Figura 2.1 a continuación.



Figura 2. 1 Modelo ZK LX-50 [<http://www.ibix.com/zkteco-lx50>]

2.1.2 Timework reloj checador

TimeWork es un sistema que permite registrar entradas y salidas a través de un teclado, o en su defecto por medio de un sensor de huella digital. Al tener cada registro, va generando reportes, gráficas de asistencia y/o puntualidad. Realiza cálculos automáticos, por ejemplo, de tiempo extra o horas laboradas, permisos, vacaciones y sanciones. Puede

llevar diferentes horarios, acceso total para recursos humanos, y fácil edición. Se puede encontrar en diferentes distribuidores como Office Depot, best buy a un aproximado de \$5,599 pesos con capacidad de 200 empleados, se puede ver la forma del reloj checador en la Figura 2.2 para dar una mayor idea de cómo se interactúa el dispositivo.



Figura 2. 2 TimeWork STA 4 [<http://www.relojchecador.com/FingerprintScanners.php>]

2.3 Las VPN

Para conectarnos a la Internet desde algún celular o una laptop generalmente se necesita una conexión con un router o modem, el cual está conectado a un proveedor de Internet que puede ser inalámbrico o por cable. Cosa diferente si se utilizan los datos de celular, ya que este trae su propio modem y se conecta desde la antena de tu teléfono, pero en realidad el sistema es el mismo, un dispositivo conectándose a otro que a fin de cuentas lleva al uso del internet desde donde necesitemos.

Una VPN (Virtual Privated Network) o red privada virtual, permite crear una red local sin que los usuarios estén físicamente conectados unos con otros, sino los conecta a través de Internet. Tiene tanto alcance como poder conectar de un país a otro del mundo como se puede ver en la Figura 2.3.

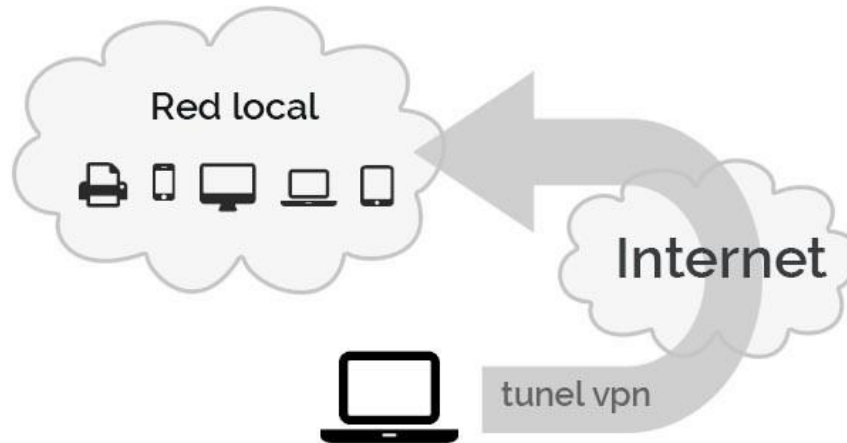


Figura 2. 3 Funcionamiento VPN [<https://www.xataka.com>]

2.2.1 VPN punto a punto

Por medio del uso de equipos exclusivos y cifrado a gran escala, cualquier empresa se puede conectar a varios sitios fijos por medio de una red pública como internet. cada equipo o sitio solo necesita una conexión a la misma red pública. Este tipo de conexión sitio a sitio también se le puede llamar intranet o extranet según su uso [3]. Por ejemplo: si se desarrolló la VPN para conectar oficinas de la misma empresa, se denomina VPN intranet. Si por el contrario la VPN se desarrolló para conectar la empresa con algún cliente o *partner*, se le denomina VPN extranet. En general hay muchos beneficios el uso de una, pero la siguiente lista son algunos de los que más interesan para este proyecto:

- Mejorar la productividad (facilita el acceso de información).
- Simplifica la topología de red.
- Posible soporte para usuarios remotos.

2.3 Introducción a las API y los Web Service

Es fácil llegar a confundirse a la hora de hablar de *API* (*Application Programming Interface*) y los *Web Service*, ya que a grandes rasgos ambos son modos de comunicación. Se puede definir al *Web Service* como aquel que facilita la interacción entre dos máquinas a través de Internet. Las *API* son una interfaz y mediante ella dos programas pueden tener comunicación. De hecho, como culmen a la introducción a este tema, se puede afirmar que no todas las *APIs* son *Web Service*, pero todos los *Web Service*, necesitan ser *APIs*. La Figura 2.4 siguiente explica el funcionamiento en general de un *web service*.

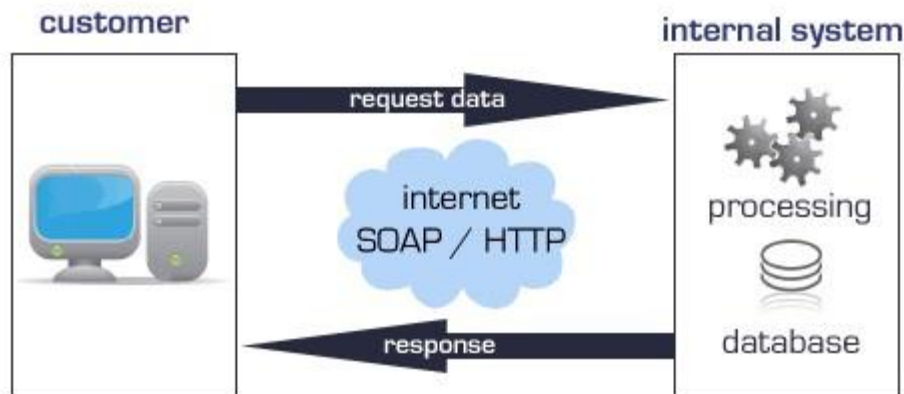


Figura 2. 4 Funcionamiento de web service

[<https://stackoverflow.com/questions/226108/what-is-a-web-service-in-plain-english>]

2.3.1 Ventajas y desventajas de utilizar Web Service y cómo funcionan.

Es posible que un *Web Service* no pueda llevar a cabo todas las tareas que una *API* podría. Esto se debe a la limitante que tiene de solo tener tres protocolos. Y cómo su nombre lo indica necesita de la web o de una conexión a Internet para funcionar.

Sin embargo, ha sido diseñado para que la interfaz se represente de un formato tal que las máquinas puedan traducirlo. Gran parte de las veces se utilizará un protocolo *HTTP* para llevar a cabo la comunicación, pero se podría emplear *SOAP*, *REST*, *XML-RPC*. Por último, pero no menos importante, el *Web Service* no tiene que contener todas las reglas que faciliten la comunicación. Esto hace que sea más fácil y rápida de realizar, a diferencia de las *APIs* [4].

La interacción entre los componentes *web* (llámese clientes, servidores o intermediarios de la *web*) dependen de la uniformidad de sus interfaces. Si cualquiera de los

componentes se desvía de los estándares establecidos, entonces la comunicación *web* no se logrará.

Los estándares establecidos se pueden explicar en los siguientes cuatro puntos:

1. Identificación de recursos. – cada concepto basado en web se tiene que direccionar con un identificador único, como un *URI (Identificador Uniforme de Recursos)*. Por ejemplo, una página particular *URI*, www.google.com, únicamente identifica el concepto de una página web y sus recursos.
2. Manipulación de recursos a través de la representación. - Los clientes manipulan la representación de los recursos, es decir, el mismo recurso puede representar a diferentes clientes de diferentes maneras. Por ejemplo, un documento podría ser representado como *HTML* en un navegador web y como *JSON* (Notación de Objetos de JavaScript, formato de intercambio de datos, fácil de interpretar para las máquinas) para un programa automatizado.
3. Mensajes Auto descriptivos. – El estado deseado de un recurso se puede representar dentro del mensaje de solicitud de un cliente. El estado actual de un recurso puede representarse dentro del mensaje de respuesta que regresa de un servidor. Como ejemplo, un cliente editor de la página wiki puede usar un mensaje de solicitud para transferir una representación que sugiera una actualización de la página (nuevo estado) desde una página web administrada por el servidor (recurso). Los mensajes autodescriptivos pueden incluir metadatos (conjunto de datos) para transmitir detalles adicionales sobre el estado del recurso, el formato y el tamaño de la representación, y el mensaje en sí. Un mensaje HTTP proporciona encabezados para organizar los distintos tipos de metadatos en campos uniformes.
4. Hipermedia como motor del estado de aplicación (HATEOAS). – Una representación del estado de recursos incluye links que relacionan recursos. Los *links* o enlaces son los eslabones que unen la web permitiendo a los usuarios manejar la información y aplicaciones de una manera útil para todos. La presencia o ausencia de un *link* en una página es una parte importante del estado actual del recurso. [4]

2.4 Herramientas de bases de datos

SQL (*Structured Query Language*) es un lenguaje para la creación, edición y consulta de bases de datos relacionales. Con él se pueden crear bases de datos, tablas, relaciones, insertar datos, modificarlos, eliminarlos, realizar consultas generales y específicas de información de las bases de datos, etc. Para gestionar una base de datos hecha con éste se requiere de un DBMS (*Database Management System*) compatible con SQL que permita administrar la organización, seguridad, entre otras cosas.

Microsoft SQL Server

Microsoft SQL Server es un DBMS desarrollado por *Microsoft* en 1989 para SQL, el cual es compatible con el sistema operativo de *Microsoft Windows* y *Linux*; y cuya versión más reciente fue lanzada en 2017 y está disponible en múltiples idiomas algunas características se pueden observar en la Tabla 1 siguiente [5] las cuales son muy popular para desarrollo web y es usado por sitios como *Facebook*, *YouTube*, *Twitter*, entre otros.

Tabla 1 Costos y compatibilidad de bases de datos. [Elaboración propia]

	Microsoft SQL Server	MySQL
Costo	Más de 1,000 dólares uso comercial. Gratis para uso propio	Precios Mayores a 2,000 dólares
Multiplataforma	Linux Windows	Windows, MacOS, Linux
Código abierto	No.	No.

2.4 Servidores

Existe un campo muy variado cuando hablamos de tipos de computadoras y a pesar de que un servidor puede ser similar a una computadora hablando de algunas partes de hardware, firmas o herramientas de software, no están dedicados para fungir de la misma manera. Como ejemplo podríamos hablar de que una computadora personal, esta diseñada para realizar tareas de oficina, utilizando algunos editores de texto, quizá programas para diseño, entre otros. A diferencia de un servidor, que está dedicado a trabajar continuamente por los servicios que brinda y que deben estar disponibles, además de que un servidor cuenta con diferentes restricciones para darle seguridad a la información que administra.

Un servidor lo podemos definir como un sistema de cómputo donde se realizan tareas que apoyan tanto a otras computadoras como aplicaciones llamadas clientes como se observa en la Figura 2.5.

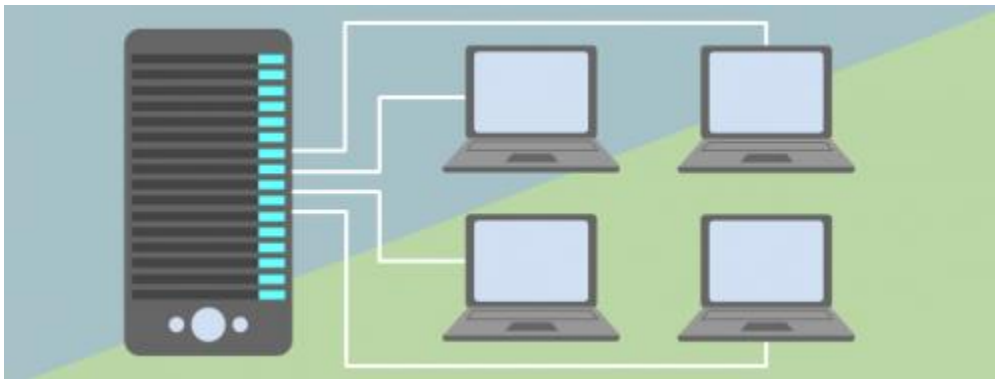


Figura 2. 5 servidor y su función [<https://www.codigohosting.com/servidor-compartido-ventajas-desventajas/>]

III. Desarrollo del proyecto

En el siguiente capítulo abarca la metodología que se utilizó para el desarrollo del sistema, los requerimientos por parte del software que se planteó instalar y qué se tuvo que hacer para que, utilizando la tecnología para la cual ya se había invertido, además se realizarán las tareas necesarias y así no se invierta más que en el desarrollo de pequeñas herramientas de software y la base de datos.

3 Metodología

Como se puede observar en la Figura 3.1 los niveles del 1 al 4, retornan del 4 al 1 proceso en la cual se valida lo que se hizo previamente. La estructura está diseñada para obedecer el principio que marca que para cada fase de desarrollo debe existir un resultado verificable. Se advierte también que la cercanía entre una fase del desarrollo y su fase de verificado va aumentando en la medida que incrementa el nivel de la V.

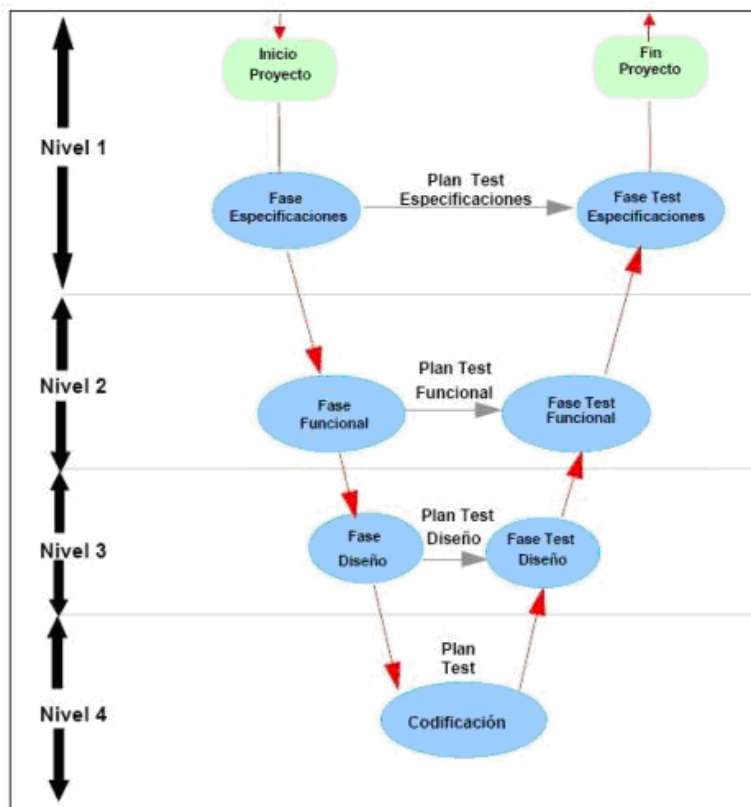


Figura 3. 1 Modelo V o de cuatro niveles
[http://ingenieriadesoftware.mex.tl/61885_Modelo-V.html]

nivel 1 a diferencia de otros niveles va orientado al cliente. Este nivel es el inicio del proyecto y el final de este. Está compuesto del análisis de requisitos y especificaciones y así se define, como fase de requisitos y especificaciones.

El nivel 2 va orientado a las funciones y características del sistema propuesto. Podría considerarse caja negra al sistema y sus características serian aquellas funciones visibles por el usuario final. Se define como documento de análisis funcional.

El nivel 3 define cada componente de hardware y software del resultado final. Se define como arquitectura del sistema.

El nivel 4 es la implementación, donde se desarrollan sus elementos o módulos del programa.

3.1 Recopilación de requisitos y especificaciones

También conocida como análisis. Esta es la fase donde se realiza la investigación a profundidad del problema que se desea solucionar, primeramente, mediante la recopilación de requisitos.

Uno de los datos de entrada o de recopilación es la solicitud donde explican al departamento de sistemas lo que necesitan para poder trabajar. A partir de esto se desarrolla un diagrama de actores que son aquellos que tendrán presencia en la funcionalidad, utilidad o uso del sistema final.

Dentro de los requisitos que nos pidió el sistema de nóminas a implementar fueron solamente tres datos como acordamos en la reunión vía telefónica envió la configuración campos de los registros de checada.

Tabla 2 Datos requeridos por el Sistema de nóminas [Elaboración propia]

EmployeeID	varchar(20)	Numero o código asociado al sistema
FechaHora	datetime	Fecha y hora de la checada
DeviceID	smallint	Numero de reloj asociado en el sistema

3.2 Documento de análisis funcional

Basados en el análisis realizado en la fase anterior:

Para la base de datos se utiliza Microsoft SQL Server, es importante señalar que la empresa Acroprint junto con TimeQplus fueron los desarrolladores de la base de datos, puesto que ellos son los proveedores de los relojes checadores, y posteriormente se estudiaron las diversas tablas para trabajar en los Qwerys del web service. Dentro de ella se cuenta con las diferentes tablas con datos de los diferentes trabajadores, con su número de trabajador, datos personales como apellido, nombre, fecha de checada, reloj en el que hubo registro *Figura 3.2*.

Aquellas tablas que se mencionan como TQ, fueron creadas para la herramienta llamada

TimeQplus que organiza todo lo referente a los registros de los relojes checadores. Las demás tablas ayudan al departamento de recursos humanos al momento de que en la semana existan días feriados, necesiten algún número de empleado, supervisores etc.

Esta base de datos estará en un servidor en la sucursal de Cd. Juárez, debido a que es la matriz, donde se reúnen el gerente con los coordinadores de servicio las demás sucursales para sus consultas y conferencias, incluso es desde donde está el departamento de sistemas, de manera que se puede dar una solución más rápida a cualquier tipo de problema físico o a nivel software del servidor.

Por último, se codificó un *Web Service*, el cual su única función será tomar una fecha que se le enviará por medio de Internet, y este responderá con las checadas de todas las sucursales en ese día, la limitante es cinco días hábiles anteriores a la última fecha laborada.

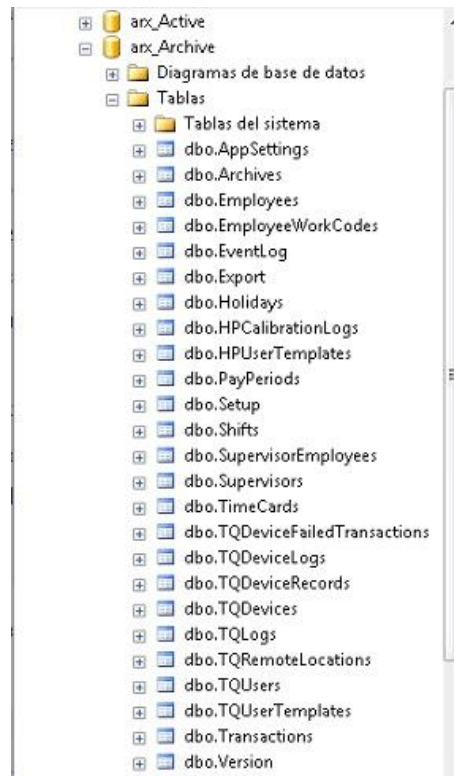


Figura 3. 2 Base de datos
[Elaboración propia]

3.3 Arquitectura del sistema

También llamada fase de diseño. Es el apartado donde se define que se utiliza para estructurar todo el proyecto completo como lo es hardware que se puede también ver en la *Figura 3.3* de forma más ejemplificada del hardware que se utilizó y que se describe a continuación.

- Servidor con Sistema operativo: |Microsoft® Windows Server® 2008 Standard versión 6.0.6002.
- Procesador: |Intel(R) Xeon(R) CPU E5-2630 0 @ 2.30GHz arquitectura: x64; 2300 MHz. Memoria RAM: |16.0 GB
- Disco Duro: |C: Local Fixed Disk; size: 800.0 GB; file system: NTFS.
- *ZKSoftware* Reloj Checador de Huella Digital y Control de Acceso.

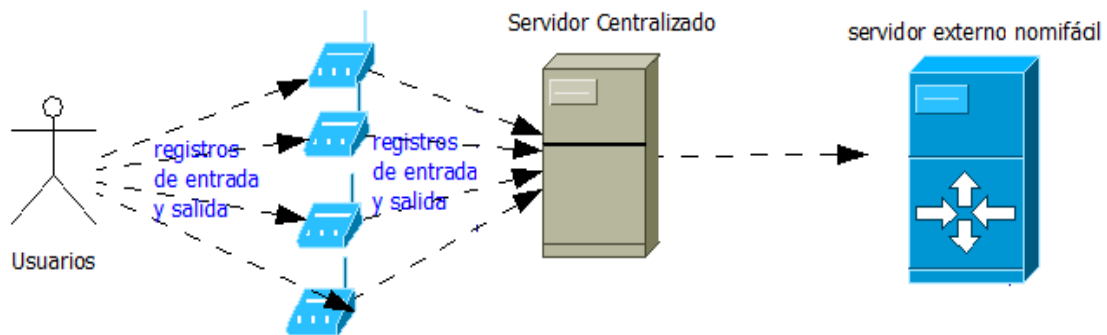


Figura 3. 3 Arquitectura Hardware [Elaboración propia]

Explicado en software se usaron los siguientes recursos:

Microsoft SQL Server 2008 para el desarrollo de la base de datos en el servidor centralizado.

TimeQPlus y *Acroprint* para la base de datos de las checadas en un servidor.

Web Service herramienta de software creada para el encapsulamiento de información solicitada por el proveedor de nomifácil.

Visual Studio 2017 lenguaje C# para codificación del web service.

3.4 Codificación

Una vez que ya se ha montado el diseño y se le ha dado forma a hardware funcional, se procedió a realizar el web Service, asignarle puerto y ponerlo en línea para hacer las pruebas de web Service en cuanto a acceso desde cualquier lugar. En la Figura 3.4 se hace referencia a detalle del funcionamiento total del sistema.

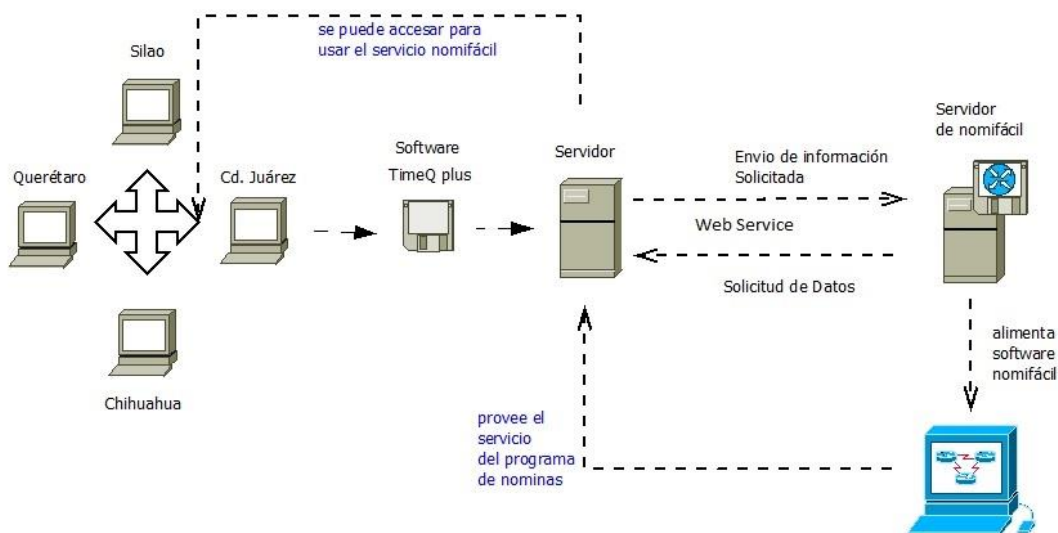


Figura 3. 4 Funcionamiento total del servidor con las herramientas de software [elaboración propia]

Por último, se creó el manual de usuario con la intención de que el proveedor al no estar familiarizado con un web Service pudiera hacer uso de este, el cual se puede encontrar en el *Anexo 1*.

IV. Resultados

El capítulo de resultados es un apartado explicando lo que se logro cada que se implementaba cada parte, este proyecto, necesitó alcanzar dos situaciones: la primera, enfocada al servidor, verificar la conexión en la red y el registro de entradas/salidas de todo GMEX y la segunda enfocada al web service ya que este se encarga de realizar búsquedas en la base de datos del servidor con información real y con ambas pruebas se verifica que corre de manera correcta.

4.1 Presentación de resultados

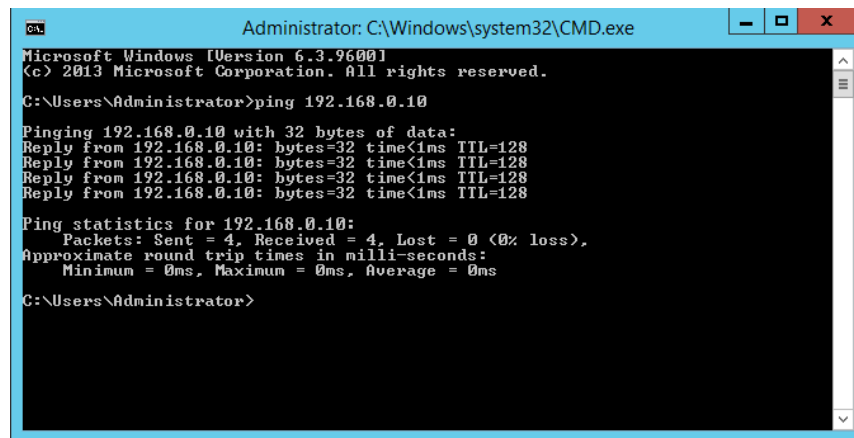
En el siguiente capítulo se abordarán las pruebas realizadas al proyecto que fueron orientados de dos maneras:

Primero, orientadas al funcionamiento de la base de datos dentro del servidor puesto que es necesario verificar que verdaderamente el servidor está siendo alcanzado en la red por los relojes, así como la base de datos dentro del servidor que esté llegando la información correctamente a las tablas.

Segundo, orientadas al web service, de manera que le damos las especificaciones del *request* que nos hacen, es decir, el *URL*, el puerto asignado, nombre del api, nombre de la función y el tipo de dato que se le debe enviar. Todo lo anterior con la finalidad de probar que está en línea, y que es segura porque solo envía la información solicitada.

4.2 Pruebas orientadas al servidor

Se verificó por medio de un ping, como se ve en la Figura 4.1, que el servidor sea alcanzado para los demás dispositivos y computadoras de la red local esto con la intención de que la información de la base de datos sea verificada por el departamento de sistemas y de recursos humanos para lo cual se asignaron perfiles con escritorios remotos.



```
Administrator: C:\Windows\system32\CMD.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128
Reply from 192.168.0.10: bytes=32 time<1ms TTL=128

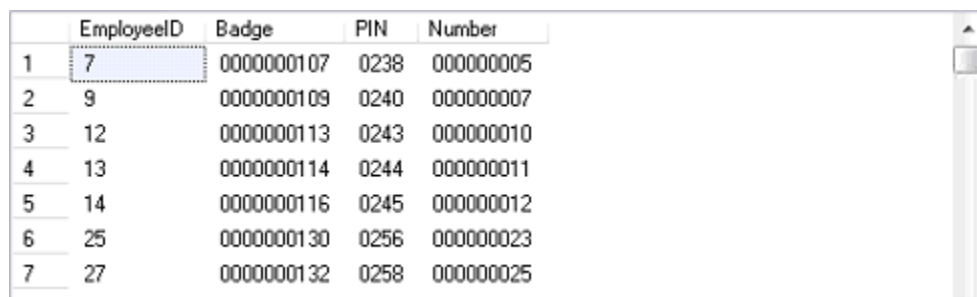
Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>
```

Figura 4. 1 Envío de Ping a ip del servidor

Se realizaron queries o búsquedas dentro de la base de datos del servidor, con el objetivo de afirmar que la información de los cuatro relojes checadores estaba centralizándose según se planteó en los objetivos específicos.

Uno de los primeros queries que se realice fue para probar la tabla de empleados, a pesar de que se cuenta con permiso por parte de la empresa para utilizar información, (Anexo 2) decidí solo mostrar la necesaria de la base de datos para las pruebas, saber que se contaba con el registro de cada empleado como lo muestro en la siguiente Figura 4.2



	EmployeeID	Badge	PIN	Number
1	7	0000000107	0238	000000005
2	9	0000000109	0240	000000007
3	12	0000000113	0243	000000010
4	13	0000000114	0244	000000011
5	14	0000000116	0245	000000012
6	25	0000000130	0256	000000023
7	27	0000000132	0258	000000025

Figura 4. 2 Tabla Empleado [Elaboración propia]

La Figura 4.3 que se muestra a continuación, no solamente es una búsqueda, sino que en ella contiene también los tres datos que se solicitaron por parte de nomifácil para poder trabajar, es importante señalar que se decidió agregar un cuarto dato con la intención de funciones futuras que podrían ser agregadas como lo son checadas remotas desde alguna aplicación de celular, pero esto se mencionará en

	TQDevLogID	TQID	EnrollNo	EmployeeID	TransactionDate	WorkCode
1...	363153	4	852	643	2019-10-12 07:43:56.000	0
1...	363154	4	893	684	2019-10-12 07:44:01.000	0
1...	363155	4	766	555	2019-10-12 07:44:48.000	0
1...	363156	4	836	627	2019-10-12 07:44:54.000	0
1...	363157	4	881	672	2019-10-12 07:44:59.000	0
1...	363158	4	702	486	2019-10-12 07:45:17.000	0
1...	363159	4	715	499	2019-10-12 07:47:19.000	0
1...	363173	3	759	548	2019-10-12 08:00:20.000	0
1...	363174	3	670	452	2019-10-12 08:00:26.000	0
1...	363175	3	821	611	2019-10-12 08:00:39.000	0
1...	363176	3	778	567	2019-10-12 08:00:44.000	0
1...	363177	3	634	416	2019-10-12 08:01:03.000	0
1...	363178	3	503	283	2019-10-12 08:02:19.000	0
1...	363179	3	818	608	2019-10-12 08:20:22.000	0
1...	363184	1	552	332	2019-10-12 07:55:50.000	0
1...	363185	1	763	552	2019-10-12 07:55:59.000	0
1...	363186	1	244	13	2019-10-12 07:56:08.000	0
1...	363187	1	404	181	2019-10-12 07:56:12.000	0
1...	363188	1	856	647	2019-10-12 07:56:15.000	0
1...	363189	1	240	9	2019-10-12 07:56:18.000	0
1...	363190	1	855	646	2019-10-12 07:56:20.000	0
1...	363210	2	744	530	2019-10-12 07:42:38.000	0
1...	363211	2	857	648	2019-10-12 07:46:24.000	0
1...	363212	2	519	299	2019-10-12 07:46:29.000	0
1...	363213	2	740	526	2019-10-12 07:46:32.000	0
1...	363214	2	853	644	2019-10-12 07:46:38.000	0
1...	363215	2	721	505	2019-10-12 07:47:07.000	0
1...	363216	2	399	175	2019-10-12 07:50:21.000	0

el apartado adecuado. Al obtener estos datos, significa que el query que se realizó es aproximado al ideal para presentar la información solicitada y que en efecto será uno de los pilares más importantes al momento de que el web service realice su función.

Figura 4. 3 Búsqueda en base de datos del servidor [Elaboración propia]

4.3 Pruebas orientadas al web service

Debido a que el servicio de nomifácil esta aun en validaciones por parte del departamento de recursos humanos y de contabilidad, no se contó con el permiso para compartir imágenes que tuvieran información de tablas, gráficas o el funcionamiento de nomifácil, sin embargo, opté por utilizar una herramienta llamada *Postman*, la cual es una extensión de un navegador el cual permite el envío de peticiones *HTTP* y *JSON* sin la necesidad de que el usuario desarrolle un cliente.

La Figura 4.4 muestra el historial de algunas de las fechas en las cuales se han realizado las pruebas utilizando *postman*, en ella se encuentra las especificaciones utilizadas, en este caso el nombre del web service con el que se empezó *GMEXRelej*.



Figura 4. 4 Historial de pruebas postman

En algunas de esas pruebas que se ven en la figura anterior se hacían pruebas solo de funcionamiento por si solo el web service, es decir que funcionara recibiendo el *GET* y dando una respuesta, más adelante como se puede observar a partir de abril 17 aproximadamente se ve un poco más extensa el *GET* que se está realizando, esto debido a que se empezaba a instalar el programa en el servidor de la base de datos, asignándole un puerto y para evitar confusiones decidí cambiarle el nombre a *GMEXWatch*. En la figura 4.5 podemos observar que clase de respuesta es la que este web service da. Una vez que se logró que la respuesta fuera correcta con los datos solicitados por

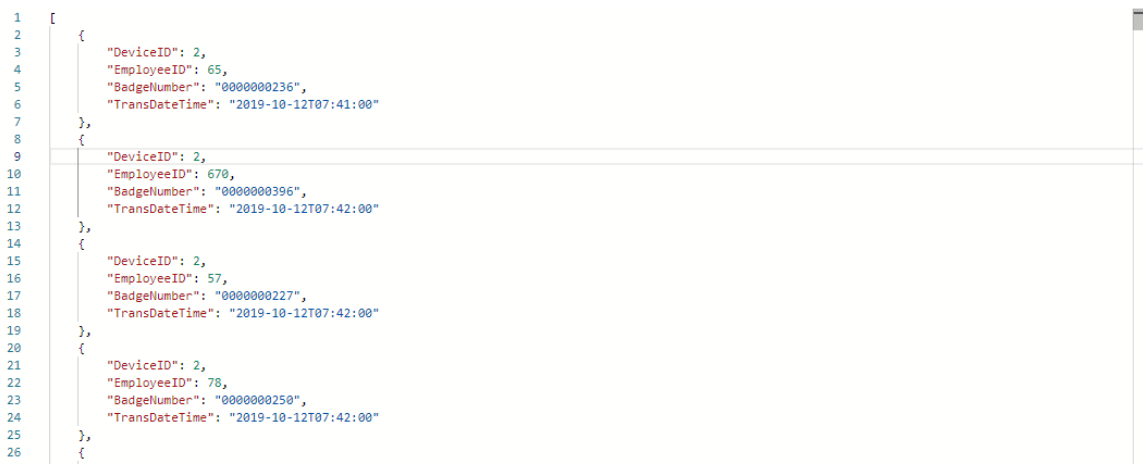


Figura 4. 5 Respuesta de web service a Postman [Elaboración Propia]

nomifácil se creó el manual de usuario disponible en el Anexo 1 de este documento, el cual cuenta con las características que debe de tener el método GET y el rango de fechas en las cuales el cliente puede solicitar la información de las checadas. Así como los posibles errores que la aplicación si se pone mal el puerto, el formato de fecha, o el nombre del web service. Logrado eso, se confirmó por correo que ya estaban ellos trabando en hacer un cliente para solicitar y descargar esa información de manera amigable para su interfaz.

V. Conclusiones

Dentro de este capítulo se aborda el último tema, una vez terminado el proyecto y las pruebas, se analiza lo que aportó el proyecto realizado, además de saber si cumplió con los objetivos planteados al inicio, así también describiremos las complicaciones que se presentaron para poder alcanzarlos.

5.1 Conclusiones del objetivo establecido

De acuerdo con las pruebas mostradas en el capítulo 3 y capítulo 4 se alcanzaron los objetivos planteados, se desglosarán uno a uno los objetivos y se explicará que parte del proyecto hizo que se alcanzara con éxito:

- Implementar un sistema de cómputo que centralice información de checadas para transferirlas y/o acceso rápido. Se alcanzó al levantar el servidor que cuenta con la base de datos y es alimentada por los relojes checadores de las diferentes sucursales según su identificador de Reloj (1,2,3,4) en la base de datos.
- Desarrollo de una herramienta web para el traspaso de información de las checadas con el objetivo de alimentar el sistema de nóminas que se requiere implementar. Se utilizó un *web service*, con el objetivo de no generar escritorio remoto a personal externo a la empresa sino poder tener un acceso controlado por el departamento de sistemas según convenga. Además, de esta manera se definirían los datos necesarios, para cumplir con el objetivo planteado de la seguridad.
- Brindar seguridad a la información compartida: se logra al momento de solicitar características específicas en la solicitud del cliente como lo son: la fecha, el puerto, el nombre del *web service*, y el nombre específico del *URL* (gmexmontacargas.com:83\api\...) mencionadas en el manual de usuario, lo cual regresa solo los datos que nomifácil previamente solicito.

Con el desarrollo del proyecto se optimizó al acceso de información de las diferentes tablas en la base de datos para corroborar datos. Segundo, se ha utilizado el servidor para realizar algunos respaldos de información de empleados y de esta manera guardarlos para su seguridad en caso de ser necesario. Tercero, se están concluyendo las pruebas finales ya de algunos procesos de nómina para implementar totalmente

nomifácil dentro de la empresa. Y cuarto, aportó velocidad a la hora de envío y recepción de datos. Cabe mencionar que ahora se hacen respaldos de la base de datos en lugar de cajas de archivo muerto lo cual ahorra gastos por rentas de almacenes de archivos en cuanto a la nómina se refiere.

5.2 Recomendaciones a futuro

En el proceso de realización del proyecto específicamente en la parte donde se estudiaban las tablas y la información de checadas de los empleados, me di cuenta que existen trabajadores de servicio que inician su jornada realizan su registro de entrada en alguna de las sucursales de la empresa, sin embargo duran toda su jornada laboral fuera de la empresa, de manera que pasada la hora de salida ellos se dirigen a sus hogares, en lugar de volver a la base a checar, y se limitan a avisar por teléfono a su supervisor de lo sucedido y éste daba aviso al departamento de recursos humanos para que hicieran lo correspondiente con el empleado. Debido a esto se está planteado la posibilidad de programar una aplicación móvil desde la cual se concedan permisos necesarios para poder checar desde donde están dando el soporte.

Referencias

- [1] L. Q. García, «<http://reini.utcv.edu.mx/handle/123456789/673>,» Universidad Tecnológica del Centro de, 1 Febrero 2016. [En línea]. Available: <http://reini.utcv.edu.mx:80/handle/123456789/673>. [Último acceso: 10 8 2019].
- [2] C. Aros Gil, «http://www.unilibre.edu.co/revistaavances/avances_10/r10_art7.pdf,» 29 Mayo 2009. [En línea]. Available: www.unilibre.edu.co. [Último acceso: 10 septiembre 2019].
- [3] C. R. Ochoa Saavedra, «Diseño e implementación de un puente inalámbrico punto multipunto para la mejora de la interconexión de las areas de la empresa plásticos rimac srl,» Chiclayo, 2012.
- [4] M. Massé, REST API Design Rulebook, United States of America: O'REILLY, 2012.
- [5] M. Corporation, «<https://www.microsoft.com/es-mx/sql-server/sql-server-2017>,» 1989. [En línea]. Available: <https://www.microsoft.com>. [Último acceso: 27 agosto 2019].

Anexos

Anexo 1

Documento de Definición de los Servicios que presta Web Service

Juan Chacón
Joel Gaspar Ramírez
Lino Andres Gallegos Garcia

Gmex Juárez, Chihuahua

Introducción: GMEXWatch es un servicio web el cual extrae información de una Base de Datos teniendo como parámetro una fecha.

Parámetros que solicita GMEXWatch:

Para presentar la información se necesita enviar como parámetro la fecha deseada en un request Get de HTML como se muestra a continuación:

<http://gmexmontacargas.com:83/api/GMEXWatch?date=2019-04-16>

Nota: El servicio brinda datos correspondientes a la semana de trabajo en curso y una semana anterior

Datos mostrados por el API:

- **DeviceID**: Numero de dispositivo (Reloj)
- **EmployeeID**: Es el secuencial de la Base de Datos (Primary Key).
- **BadgeNumber**: Es el número de empleado y gafete.
- **TransDateTime**: Fecha y hora de la transacción (Checada).

**Los datos con marca amarilla son los solicitados por el Usuario.

```
1- [
2- {
3-   "DeviceID": 1,
4-   "EmployeeID": 552,
5-   "BadgeNumber": "0000002054",
6-   "TransDateTime": "2019-04-13T07:27:00"
7- },
8- {
9-   "DeviceID": 1,
10-  "EmployeeID": 533,
11-  "BadgeNumber": "0000002047",
12-  "TransDateTime": "2019-04-13T07:34:00"
13- },
14- {
15-   "DeviceID": 2,
16-   "EmployeeID": 78,
17-   "BadgeNumber": "000000250",
18-   "TransDateTime": "2019-04-13T07:34:00"
19- },
20- {
21-   "DeviceID": 2,
22-   "EmployeeID": 36,
23-   "BadgeNumber": "000000203",
24-   "TransDateTime": "2019-04-13T07:35:00"
25- },
26- {
27-   "DeviceID": 2,
28-   "EmployeeID": 599,
29-   "BadgeNumber": "000000376",
30-   "TransDateTime": "2019-04-13T07:39:00"
```

Si no se encuentran datos o de no cumplirse con el parámetro de manera correcta, aparecerán el siguiente error:

```
1  {
2    "Message": "La solicitud no es válida.",
3    "MessageDetail": "El diccionario de parámetros contiene una entrada NULL para el parámetro 'date' del tipo que no acepta valores NULL 'System
4    .DateTime' del método 'System.Data.DataTable Get(System.DateTime)' en 'WebServiceReioj.Controllers.SearchController'. Un parámetro opcional
    debe ser un tipo de referencia, un tipo que acepte valores NULL o debe declararse como parámetro opcional."
  }
```



11 de Octubre de 2019

A quien corresponda.

Por medio del presente, la empresa Grupo Maquirrental Express S.A. de C.V. autoriza a Lino Andres Gallegos Garcia hacer uso del proyecto denominado "GmexWatch" para ser presentado como proyecto de titulación; mismo que él desarrolló y colaboró en la implementación. Dicho proyecto surgió a raíz de las necesidades de los nuevos procesos de nóminas en la empresa.

Autoriza: Juan Manuel Chacón

Analista en sistemas



📍 **CHIHUAHUA**
Heroico Colegio Militar 5504
Col. Nombre de Dios
Chihuahua, Chih. C.P. 31105
☎ (614) 424-3333

📍 **JUÁREZ**
Melón 6834
Col. El Granjero
Juárez, Chih. C.P. 32690
☎ (656) 611-3333

📍 **QUERÉTARO**
Acceso 1 122
Fracc. Industrial Montaña
Querétaro, Qro. C.P. 76150
☎ (442) 216-3000

📍 **SILAO**
Circuito Las Vías 602
Col. Bustamante
Silao, Gto. C.P. 36128
☎ (472) 722-1100