

UNIVERSIDAD DE SONORA
DIVISIÓN DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

REPORTE DE PRÁCTICAS PROFESIONALES

**CONTROL Y MONITOREO DE USUARIOS
Y SERVIDORES**

PRESENTA:

JOSÉ ISMAEL SILVAS RODRÍGUEZ

INGENIERÍA EN SISTEMAS DE INFORMACIÓN

TUTOR:

DR. FEDERICO MIGUEL CIRETT GALÁN

ÍNDICE GENERAL

1 INTRODUCCIÓN.....	5
2 JUSTIFICACIÓN.....	6
3 OBJETIVOS.....	6
4 DESCRIPCIÓN DEL CONTEXTO	7
4.1 Entorno donde se ubica la unidad receptora	7
4.2 Equipamiento e instalaciones	8
4.3 Normatividad de la unidad receptora	8
5 FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS	9
5.1 Linux	9
5.2 BASH	9
5.3 Script.....	9
5.4 Vim.....	9
5.5 Apache Subversion	9
5.6 Puppet.....	9
5.7 Máquina Virtual (VM).....	10
5.8 Hypervisor	10
5.9 GitLab	10
5.10 Nagios.....	10
5.11 Jira	10
5.12 Slack	11
5.13 Internet Protocol (IP)	11
5.14 Domain Name System (DNS).....	11
5.15 Secure Shell (SSH)	11
5.16 Llave SSH	12
5.17 Password Hash (passhash).....	12
6 PROCEDIMIENTOS EMPLEADOS Y ACTIVIDADES DESARROLLADAS.....	13
6.1 Creación de cuentas en Puppet	13
6.2 Creación de cuentas en el repositorio svn.....	14
6.3 Creación de cuentas en GitLab	15
6.4 Agregar usuarios a grupos en GitLab	17
6.5 Eliminar y bloquear cuentas de usuarios	18

6.6 Acceso de usuarios a VMs	20
6.7 Añadir y modificar registros DNS.....	21
6.8 Añadir y actualizar paquetes en VMs	23
6.9 Hosts en Nagios	24
7 RESULTADOS OBTENIDOS.....	26
8 CONCLUSIONES Y RECOMENDACIONES	27
9 REFERENCIAS	28

ÍNDICE DE FIGURAS

Figura 1 – Logo de la empresa	7
Figura 2 – Ubicación de la empresa.....	7
Figura 3 – Espacio de trabajo	8
Figura 4 – Interfaz de GitLab para crear un usuario, parte 1	16
Figura 5 – Interfaz de GitLab para crear un usuario, parte 2.....	16
Figura 6 – Interfaz de GitLab para agregar un usuario a un grupo.....	17
Figura 7 – Interfaz de GitLab para bloquear un usuario	19

1 INTRODUCCIÓN

En el reglamento de la Universidad de Sonora se contempla que en todos los planes de estudio se incluyan actividades de vinculación con el sector social o productivo con el propósito de complementar la formación de los estudiantes a través de la aplicación de los conocimientos obtenidos en las diversas materias de las carreras, y a su vez, desenvolvemos en el ambiente laboral que nos espera una vez finalizados nuestros estudios. Así, la carrera de Ingeniería en Sistemas de Información, del Departamento de Ingeniería Industrial y de Sistemas, incluye en su plan de estudios las prácticas profesionales con un valor de 20 créditos, que son equivalentes a 340 horas.

Con el objetivo de cumplir con los créditos requeridos en el plan de estudios de mi carrera por concepto de prácticas profesionales, me decidí a buscar una institución donde pudiera aplicar y expandir los conocimientos adquiridos a lo largo de mis estudios profesionales. Así, llegué a Internet brands, que, por medio de Intugo, me dieron la oportunidad de realizar mis prácticas profesionales con ellos en el área de Network Operations Center (NOC).

En el presente documento se describen las actividades que desarrollé en el transcurso de mis prácticas profesionales en Internet Brands, así como, una breve introducción de la empresa, el fundamento teórico para entrar en contexto, en particular, sobre las herramientas que se utilizan en la empresa y los conocimientos adquiridos a lo largo de las mismas.

2 JUSTIFICACIÓN

Llevar un control de la creación, eliminación y modificación de la información de usuarios, proveer el acceso a los servidores que requieren, así como los permisos en carpetas y archivos necesarios, tanto como los comandos que pueden ejecutar. También proveer acceso al repositorio SVN en los archivos requeridos por el usuario. Además de la creación, bloqueo de usuarios y control de acceso de los mismos a los grupos y proyectos de trabajo en GitLab.

Por otro lado, es necesario que los dispositivos en la red estén monitoreados las 24 horas del día, esto se lleva a cabo mediante Nagios, la herramienta de monitoreo de red utilizada por la empresa, así se pueden observar y resolver los problemas en tiempo real para evitar que estos se prolonguen más del tiempo debido.

3 OBJETIVOS

Capacitar de manera íntegra al practicante en las herramientas que se utilizan en el área de Network Operations Center (NOC) dentro de la empresa, inicialmente familiarizándose con las actividades descritas en el punto anterior y posteriormente manejar el resto de actividades del NOC para que, una vez terminadas las prácticas profesionales, pase a formar parte del equipo.

4 DESCRIPCIÓN DEL CONTEXTO

La empresa fue fundada originalmente en 1998 bajo el nombre de CarsDirect.com, en ese entonces se enfocaba en administrar páginas en el ámbito automotriz. En el año 2006 la empresa adquiere el nombre de Internet Brands y de esta manera amplía su mercado hacia otras áreas. Actualmente se enfoca en 4 principales ramas: Automotriz, Salud, Jurídico y Hogar/Viajes, entre otras, en menor medida.

El enfoque por verticales de Internet Brands combina soluciones web líderes para empresas y sitios web para consumidores, así logra trabajar con todo tipo de clientes. En la actualidad, Internet Brands tiene más de 100,000 clientes alrededor del mundo que suman más de 250 millones de visitas al mes.



Figura 1 – Logo de la empresa

4.1 Entorno donde se ubica la unidad receptora

En Hermosillo, Internet Brands se ubica dentro de la empresa Intugo en Plaza Girasol, con dirección Blvd. Solidaridad #56 local #6, colonia Sahuaro.

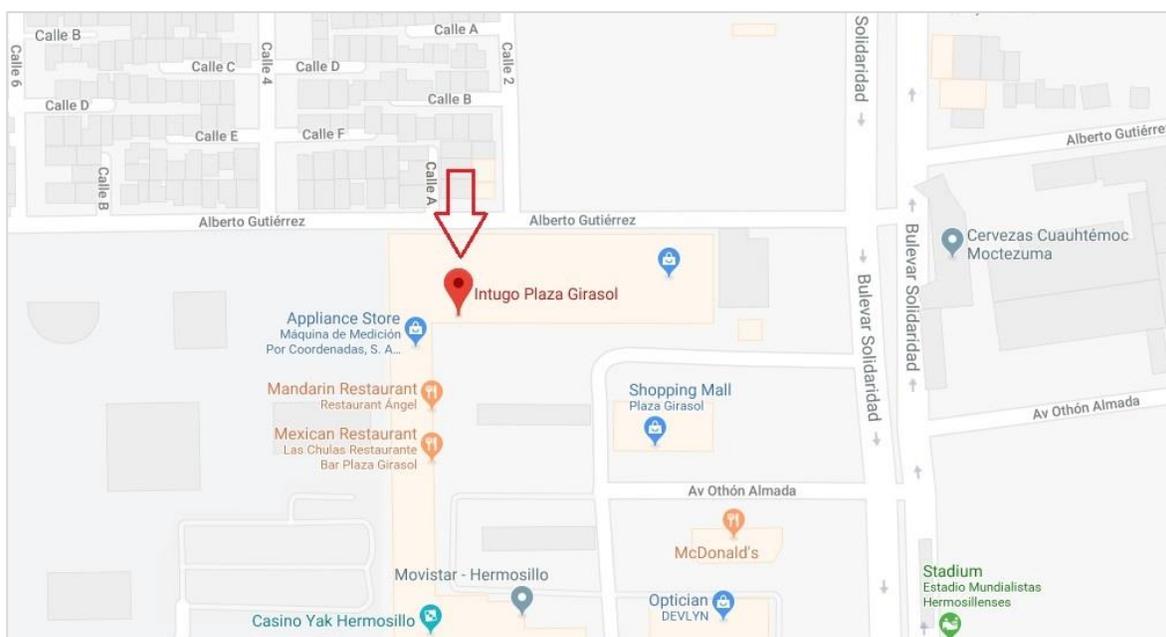


Figura 2 – Ubicación de la empresa

4.2 Equipamiento e instalaciones

Las instalaciones de la empresa están bien equipadas, toda la oficina se encuentra refrigerada, cada empleado en el área de NOC cuenta con un escritorio y una silla de oficina, una computadora con dos monitores, unos audífonos con micrófono (headsets), una cámara web, varias salas de junta, área de impresión, control de acceso mediante tarjeta, vigilancia mediante cámaras las 24 horas, baños, lockers, área de descanso, comedor y área de cocina.

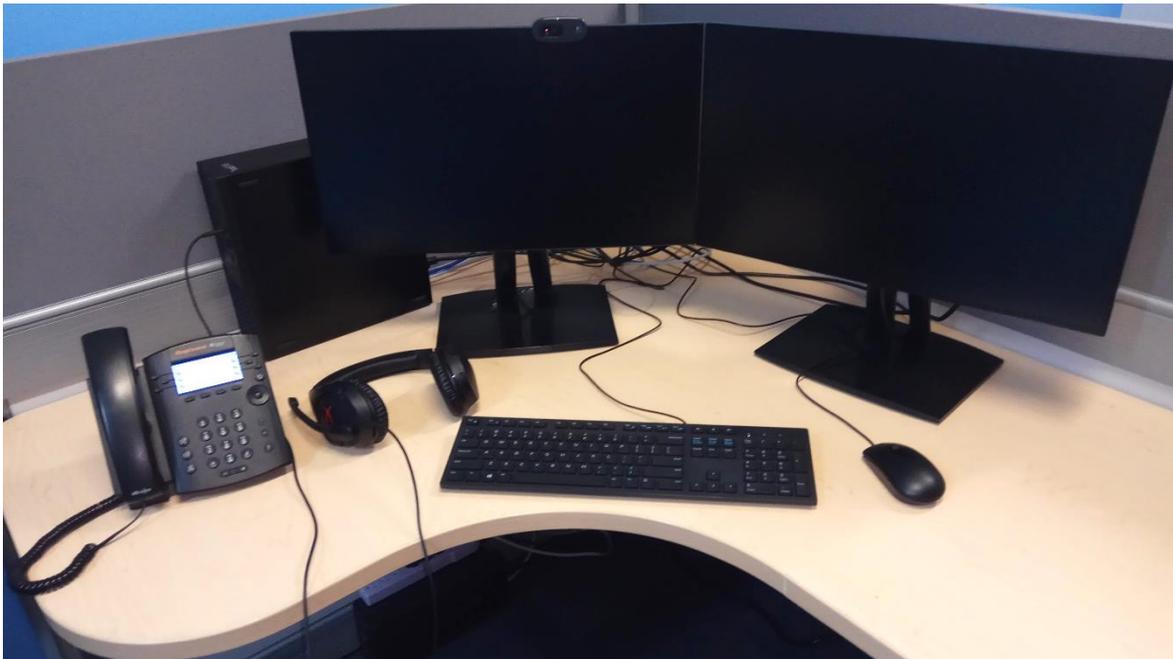


Figura 3 – Espacio de trabajo

4.3 Normatividad de la unidad receptora

Dentro de la empresa existen una gran cantidad de normas que cumplir, como practicante, las más importantes que debo cumplir son:

- Puntualidad y asistencia, se deben acumular 30 horas de trabajo a la semana, cinco días de seis horas, en mi caso.
- Siempre se debe portar el gafete de identificación con la tarjeta de acceso visible mientras se permanezca dentro de la empresa.

5 FUNDAMENTO TEÓRICO DE LAS HERRAMIENTAS Y CONOCIMIENTOS APLICADOS

5.1 Linux

Es el sistema operativo de código abierto más popular de los últimos años, originalmente desarrollado por Linus Torvalds, fue lanzado al público en marzo de 1994 y mejorado y personalizado por la comunidad con el paso del tiempo.

5.2 BASH

Es un lenguaje de programación soportado en la línea de comandos de Linux y uno de los más populares actualmente. Después su lanzamiento, en 1989, ha sido ampliamente distribuido en varias versiones de Linux.

5.3 Script

Es un archivo ejecutable que contiene un conjunto de comandos que se ejecutan en el orden que están escritos.

5.4 Vim

Es un editor de texto configurable, permite moverte con alta eficiencia en archivos de texto. Es un clon del editor de textos Vi, pero a diferencia de este, posee muchas mejoras para agilizar aún más el trabajo.

5.5 Apache Subversion

Normalmente abreviado SVN, por su comando central *svn*, es un sistema de control de versiones de código abierto desarrollado por Apache Software Foundation en octubre del año 2000. SVN provee a los usuarios una manera ordenada de trabajar, mediante la cual, todos tienen la versión más actualizada de los archivos de trabajo.

5.6 Puppet

Es una herramienta de código abierto de administración de configuración de software lanzada en el año 2005 por Luke Kanies. Permite la automatización de la configuración en las máquinas virtuales y hypervisors.

5.7 Máquina Virtual (VM)

Es un programa que actúa como una computadora virtual, se ejecuta sobre el sistema operativo instalado en el dispositivo físico, conocido como *host*, (en este caso un Hypervisor). Una VM utiliza los recursos físicos del host para poder ejecutarse.

5.8 Hypervisor

También conocido como monitor de máquinas virtuales, es un software o hardware que crea y ejecuta máquinas virtuales (VMs). Permite la ejecución de varias VMs, pueden ser con el mismo o diferente sistema operativo, en el mismo equipo y de manera síncrona, compartiendo recursos y hardware para aprovecharlos al máximo.

5.9 GitLab

Es un administrador de repositorios Git basado online lanzado en 2011 por GitLab Inc. Permite a los desarrolladores, administradores de bases de datos, aseguramiento de la calidad, etc., administrar su código mediante grupos de trabajo, y en estos, separado por proyectos.

5.10 Nagios

Es una herramienta de monitoreo de red de código abierto lanzada en marzo del 2002. Permite analizar la red en busca de problemas en dispositivos de red, ya sean sobrecarga de procesos, tráfico de red, uso de disco de almacenamiento, respuesta a ciertas variables que se configuran manualmente, etc.

5.11 Jira

Es un producto software desarrollado por Atlassian, Inc. en 2002. Provee al usuario reporte y seguimiento de problemas y necesidades, mediante *tickets*; administración por proyectos, historial de reportes, etc.

Al momento de que se crea un ticket en Jira, se debe asignar a una persona para que empiece a trabajar en él, cualquier persona que sea parte del proyecto donde se encuentra creado el ticket puede asignarlo a una persona.

En Jira, un ticket, puede tener varios estados, entre los principales están:

- **Abierto**, no hay una persona que esté trabajando en él.
- **En proceso**, actualmente se está trabajando en él.
- **En espera**, el ticket necesita un trabajo previo para ser tratado o, el cliente o la persona asignada pospuso la fecha de trabajo en él.
- **Más información requerida**, el ticket no describe el problema o la situación de manera completa, por lo que no puede empezar a trabajar en él.
- **Cerrado**, problema descrito en el ticket fue resuelto o, por alguna razón, no es posible realizar lo solicitado.

Así mismo en un ticket se pueden hacer comentarios, adjuntar URLs, archivos e imágenes para explicar mejor el problema o situación.

5.12 Slack

Es un espacio de trabajo que conecta a todas las personas dentro de una organización, donde pueden comunicarse, compartir documentos, código, etc. Se pueden crear espacios de trabajo privados, llamados canales.

5.13 Internet Protocol (IP)

Es un conjunto de reglas que permite a los dispositivos, conectados a internet, enviar y recibir la información entre sí.

5.14 Domain Name System (DNS)

Los humanos buscamos información en internet mediante nombres de dominio, pero los navegadores web se comunican a través de direcciones IP. La función del DNS es traducir los nombres de dominio a direcciones IP para que el navegador pueda cargar las páginas solicitadas.

5.15 Secure Shell (SSH)

Es un protocolo que utiliza encriptación para permitir la conexión segura entre dos dispositivos conectados en una red.

5.16 Llave SSH

Es la credencial de acceso que utiliza el protocolo SSH. Utiliza dos llaves, una pública y una privada, la llave privada es la llave personal, no debe ser compartida con nadie, la llave pública es una llave que se puede compartir para dar acceso a diferentes dispositivos y sistemas.

5.17 Password Hash (passhash)

Es una forma de encriptación de contraseñas de usuarios en sistemas con control de acceso. Evita que alguien no autorizado en el sistema obtenga las contraseñas de los usuarios.

6 PROCEDIMIENTOS EMPLEADOS Y ACTIVIDADES DESARROLLADAS

6.1 Creación de cuentas en Puppet

Las cuentas de usuarios en Puppet, sirven para dar acceso a los usuarios (desarrolladores de software, administradores de bases de datos, administradores de sistemas, etc.) a las VMs.

Este proceso comienza cuando se crea un ticket en Jira, en el cual especifican el o los usuarios que necesitan cuenta en Puppet, el usuario o su Administrador de Proyecto (PM) es quién hace la solicitud de la nueva cuenta. En el ticket se debe especificar el nombre, el correo electrónico, el passhash y la llave ssh pública del usuario que necesita la cuenta.

Es común que no provean el passhash y/o la llave ssh pública, en ese caso, se hace un comentario en el ticket donde se le da la siguiente información al usuario:

- Se le indica la dirección web institucional en donde crear su passhash.
- Se le indica la dirección web en donde se describen los pasos a seguir para crear sus llaves ssh (en caso de no saber cómo crearla).

Una vez el usuario provea la información requerida, se procede a crear el usuario y cambiar el estado del ticket a “en proceso”.

Primeramente, se debe de actualizar la versión del repositorio svn en la máquina local para que no exista una discordancia de la información más actual, esto se logra con el comando `svn up`.

**Para futuras referencias en este documento, texto en *cursiva* significa el valor de la variable y texto entre corchetes [] significan argumentos opcionales del comando. Como ejemplo de nombre de usuario en los siguientes ejemplos utilizaré el nombre de usuario de mi cuenta en Puppet.

Un usuario en Puppet se crea mediante un script, desarrollado por miembros anteriores del equipo, llamado `ppeditor.sh`. Utilizando las opciones `-nu`, como se muestra en el siguiente ejemplo:

```
./ppeditor.sh -nu isilvas [-T tipo_usuario]
```

El nombre de usuario es correspondiente a la primera letra del nombre de la persona seguido de su primer apellido, en caso de que el nombre de usuario ya exista se toman la primera y segunda letra del nombre seguido de su primer apellido, y así sucesivamente.

El script anterior, con las opciones `-nu`, crea dos archivos, el primero es el archivo `isilvas.pp` (que se agrega a la carpeta donde se encuentran las cuentas Puppet),

este es el archivo de la configuración de la cuenta del usuario en el que se almacena principalmente, el nombre, correo electrónico, número de usuario y passhash; el segundo es el archivo *isilvas-sshkeys* (que se agrega a la carpeta donde se encuentran las llaves ssh de los usuarios), en este archivo se almacena la llave ssh pública del usuario, además, el script, actualiza un archivo llamado *userlist* (este archivo es solamente para mantener un registro de todos los usuarios que han trabajado en la empresa), con el nombre y número de usuario y nombre de la persona..

Una vez ejecutado el script *ppeditor.sh*, se ejecuta el comando: `svn diff`.

Este comando muestra los cambios hechos en los archivos dentro del repositorio *svn*, así se puede observar y verificar que no haya algún error de sintaxis en alguno.

Después se procede a enviar un correo electrónico, que es recibido por el equipo de NOC y Unix Admin. En este correo se copia el texto que muestra el comando `svn diff` para que alguien más revise la sintaxis, ya sea que falte información o algún otro error que haya pasado por alto. Si se encuentra un error en el correo, se hace saber mediante una respuesta del mismo, se corrige el error y se le da respuesta al correo enviado originalmente para ser revisado de nuevo. Una vez que este correo es respondido con una aprobación se ejecuta el siguiente comando:

```
svn commit -m "cambios y razón"
```

Este comando guarda la nueva versión del o de los archivos modificados en el repositorio, y así todos podrán ver el cambio una vez actualicen la versión de *svn* en su máquina local (con el comando `svn up`).

Por último, en Jira, se le informa al usuario que su cuenta en Puppet ha sido creada con la información proporcionada y se procede a cerrar el ticket.

6.2 Creación de cuentas en el repositorio *svn*

Para este proceso que el usuario cree un ticket en Jira, donde él o su PM pide acceso a una o varias secciones del repositorio *svn*.

Si el usuario no tiene cuenta en Puppet, se requiere que el usuario proporcione el passhash (al igual que en las cuentas en Puppet). En caso de que no lo proporcione, se hace un comentario en el ticket indicando la dirección web en donde crearlo.

Por otro lado, si el usuario ya tiene cuenta en Puppet, se toma el nombre de usuario de esa cuenta y se copia el passhash del archivo `isilvas.pp` para usarlos en la cuenta del repositorio svn.

El nombre de usuario de las cuentas en el repositorio svn sigue el mismo formato de las cuentas en Puppet (la primera letra del nombre de la persona seguido de su primer apellido).

Los permisos para las cuentas del repositorio svn son:

- r: permiso de lectura, permite ver los archivos en la sección solicitada.
- w: permiso de escritura, permite editar los archivos en la sección solicitada.

Antes de crear la cuenta en el repositorio svn, se actualiza el repositorio a la versión más nueva con el comando `svn up`, y se procede a agregar su nombre de usuario a la sección del repositorio solicitada en el archivo `accessfile` con sus respectivos permisos, y el nombre de usuario y passhash al archivo `passwdfile`.

Una vez agregado el usuario y su passhash a los archivos necesarios, se ejecuta el comando `svn diff` para ver los cambios realizados.

A diferencia de las cuentas en Puppet, no es necesario enviar un correo de aprobación, solo se procede a guardar los cambios en el repositorio svn mediante el comando `svn commit -m "cambios y razón"`.

Finalmente, se cierra el ticket y se le informa al usuario que su cuenta en el repositorio svn fue creada y tendrá acceso a la sección o secciones solicitadas dentro del mismo con los permisos requeridos.

6.3 Creación de cuentas en GitLab

Este proceso empieza cuando se crea un ticket en Jira, donde el usuario o su PM solicita la creación de una cuenta en GitLab.

Para este proceso se necesita solo de su nombre y correo electrónico, ya que el nombre de usuario sigue el mismo formato que las cuentas en Puppet y svn. Para establecer la contraseña de la cuenta se envía una URL por correo electrónico para

que el usuario la establezca. Las siguientes imágenes describen los pasos para crear una cuenta en GitLab:

New user

Account

Name * required 1

Username * required 2

Email * required 3

Password

Password **Reset link will be generated and sent to the user.
User will be forced to set the password on first sign in.**

Figura 4 – Interfaz de GitLab para crear un usuario, parte 1

Access

Projects limit 4

Can create group

Access level Regular 5
Regular users have access to their groups and projects

Admin
Administrators have access to all groups, projects and users and can manage all features in this installation

External
External users cannot see internal or private projects unless access is explicitly granted. Also, external users cannot create projects or groups.

6

Figura 5 – Interfaz de GitLab para crear un usuario, parte 2

1. Escribir el nombre de la persona.
2. Escribir el nombre de usuario.
3. Escribir el correo electrónico.
4. Establecer en 0 el número límite de proyectos.
5. Seleccionar el tipo de usuario regular.

6. Presionar en el botón crear usuario.

Una vez hecho esto, se comenta en el ticket para informarle al usuario que la cuenta para fue creada y se ha enviado un correo electrónico con instrucciones para establecer la contraseña de la cuenta. Finalmente, se cierra el ticket.

6.4 Agregar usuarios a grupos en GitLab

Esta tarea puede ser solicitada mediante un ticket en Jira o en el canal de slack de NOC, donde el usuario o su PM solicita acceso a un grupo de trabajo en GitLab.

En GitLab, existen 4 tipos de permisos para los grupos, los cuales son:

- Invitado.
- Desarrollador.
- Master.
- Dueño.

The screenshot displays the GitLab interface for managing a group. At the top, there is a search bar for the group name, labeled "Nombre del grupo", and an "Edit" button. Below this, the "Group info" section provides details such as the group name, path, description, visibility level (Internal), creation date (Apr 17, 2014 2:03pm), storage usage (1.1 GB), and Git LFS status (Enabled for all projects). To the right, the "Add user(s) to the group" section includes a link to project permissions, a search input field (labeled 1), a dropdown menu showing "Developer" (labeled 2), and an "Add users to group" button (labeled 3). Below these sections, the "Miembros del grupo" (Group members) section shows a list of 15 members with their roles (Owner or Master) and when they joined. At the bottom left, a "Proyectos" (Projects) section shows a list of projects with their sizes.

Figura 6 – Interfaz de GitLab para agregar un usuario a un grupo

Este es un proceso sencillo, que se describe a continuación.

1. Escribir el nombre de la persona o nombre de usuario.
2. Seleccionar el tipo de permisos que tendrá el usuario en el grupo.
3. Presionar en el botón añadir el usuario al grupo.

Al finalizar esto, se hace un comentario en el ticket informando al usuario que ha sido agregado al grupo solicitado con los permisos requeridos y se procede a cerrar el ticket o se le comenta por slack a la persona que ha sido añadida al grupo solicitado en GitLab.

6.5 Eliminar y bloquear cuentas de usuarios

Esto se inicia desde que recursos humanos envía un correo electrónico indicando que un usuario ha firmado su renuncia con la empresa. Se revisa si el usuario tiene cuenta en Puppet, svn y/o GitLab, de ser así se procede a crear un ticket en Jira para empezar el proceso de eliminar la(s) cuenta(s) del usuario.

Si el usuario tiene cuenta en Puppet o en el repositorio svn, primeramente, se tiene que actualizar la versión del repositorio en la máquina local con el comando `svn up`. Una vez actualizado, se ejecuta el script `ppeditor.sh` con las opciones `-tu`, como se muestra a continuación:

```
./ppeditor.sh -tu isilvas
```

El script `ppeditor.sh` con las opciones `-tu` elimina el archivo `isilvas-sshkeys`, actualiza el archivo `isilvas.pp` actualizando el valor de una variable que indica que el usuario ya no es empleado de la empresa, de esta manera Puppet distribuirá los cambios a todas las VMs y la cuenta se eliminará las mismas, también se actualiza el archivo `userlist` agregando al final del nombre de usuario la fecha de renuncia.

Si el usuario tiene cuenta en el repositorio svn, el mismo script elimina el nombre de usuario del archivo `accessfile` y su passhash del archivo `passwdfile`.

Una vez hecho esto, se ejecuta el comando `svn diff` para ver los cambios, estos se envían por correo para aprobación. Una vez aprobado el cambio se ejecuta el comando `svn commit -m "cambios y razón"`.

Por otro lado, si el usuario tiene cuenta en GitLab, se procede a entrar al portal de GitLab y bloquear la cuenta como se muestra a continuación.

The screenshot shows the GitLab user management interface. On the left, there is a user profile card with a red box labeled '1' around the user's name and another red box labeled '2' around the 'Profile page' link. Below this is a detailed account information section. On the right, there are two main action panels. The top panel, titled 'Block this user', lists the effects of blocking a user and includes a red button labeled 'Block user' with a red box labeled '3' around it. The bottom panel, titled 'Remove user', lists the effects of deleting a user and includes a red button labeled 'Remove user'. Below that is another red button labeled 'Remove user and contributions'.

Figura 7 – Interfaz de GitLab para bloquear un usuario

1. Nombre del empleado.
2. Nombre de usuario.
3. Botón para bloquear la cuenta.

Esto es con el fin de mantener un registro de los usuarios en la empresa y en dado caso que el usuario regrese a la empresa en un futuro, la cuenta de activa de nuevo.

Al finalizar lo anterior, se cierra el ticket y se indican los tipos de cuentas que fueron eliminadas y bloqueadas.

6.6 Acceso de usuarios a VMs

Este proceso comienza cuando se crea un ticket en Jira, donde el usuario o su PM solicita acceso a una o varias VMs. Si el usuario es el que solicita acceso a VMs se necesita aprobación del PM antes de poder dar el acceso, para lo cual se hace un comentario en el ticket pidiendo la aprobación. Por otro lado, si el PM es el que solicitó el acceso para el usuario se empieza a trabajar de inmediato.

Antes de empezar a trabajar se actualiza la versión del repositorio en la máquina local con el comando `svn up`.

Para comenzar con este proceso se realizan los siguientes pasos:

1. Se abre un archivo llamado `nodes.pp`, donde se definen todas las VMs que existen en la empresa.
2. Dentro de este archivo se busca donde está definida la o las VMs solicitadas.
3. En la definición de la VM se toma el valor de una variable llamada `users`. Este valor, también es el nombre de un archivo con extensión `.pp`
4. Se busca y abre el archivo mencionado (dentro de la carpeta donde se encuentran los archivos de acceso en `svn`).
5. Se agrega el nombre de usuario al archivo.
6. Se repiten los pasos anteriores para cada una de las VMs solicitadas, el archivo donde se agregan los usuarios, normalmente es nombrado por el grupo de trabajo que trabaja en esas VMs, por lo que muchas VMs utilizan el mismo archivo para dar acceso a usuarios, en este caso solo habría que añadir el usuario a un archivo.

Una vez terminado con todas las VMs solicitadas, se ejecuta el comando `svn diff` y se envía un correo electrónico con el resultado del comando anterior para que pase por aprobación.

Una vez aprobados los cambios, se ejecuta el comando `svn commit -m "cambios y razón"`.

Se ejecuta un script llamado `puppetsvnupdate`, que actualiza la configuración de Puppet en los servidores a la más reciente.

Después se comenta en el ticket que el acceso ha sido concedido a las VMs solicitadas y ya puede acceder a ellas. Y finalmente se cierra el ticket.

6.7 Añadir y modificar registros DNS

Esta tarea puede ser solicitada mediante un ticket en Jira o mediante un correo electrónico solicitando nuevos registros DNS o actualizar los ya existentes.

Los registros DNS son archivos que le indican al servidor DNS el valor que está asociado a un subdominio. Existen varios tipos de registros, entre los más comunes están los siguientes:

- **A**, indica una dirección IP.
- **CNAME**, relaciona un nombre alias con un nombre de dominio.
- **MX**, dirigen el correo electrónico de un dominio a los servidores del dominio.
- **TXT**, proporciona información de texto a fuentes externas del dominio.

Además de esto, existe una variable llamada TTL (tiempo de vida, por sus siglas en inglés), le indica el tiempo en segundos al servidor DNS para que se apliquen los cambios hechos.

Por ejemplo, el dominio `ejemplo.com` está asociado a la IP `72.52.4.119`, supongamos que queremos agregar el registro `otro` al dominio `ejemplo.com` y que apunte a la IP `72.52.4.120`, siendo este subdominio `otro.ejemplo.com`.

Para lograr, primero tenemos que verificar que la página web esté administrada por Internetbrands, para hacer esto se ejecuta el siguiente comando:

```
dig -tns ejemplo.com [+short]
```

Este comando muestra un resultado con los Name Server (NS) donde se puede observar si el dominio está administrado por la compañía o no. De ser positivo, procedemos a trabajar.

Por cada dominio de la compañía existe un registro DNS llamado `pri.ejemplo.com` y para esto se realizan los siguientes pasos:

1. En el repositorio `svn`, dentro de la carpeta donde se encuentran todos los dominios, se busca el archivo nombrado por el dominio y se abre.
2. Primeramente, se debe editar el serial, este se encuentra en la parte superior del archivo y es la manera en la que el DNS sabe que se ha hecho un cambio en los registros. El formato del serial es una fecha con dos números adicionales que indican el número de cambio en el día, por ejemplo `2018121702` indica que el último cambio fue el 17 de diciembre del 2018 y es el tercer cambio del día (ya que empieza desde 0).
3. Si es un nuevo registro, se añade, en orden alfabético para mantener un orden, bajo el siguiente formato:

```
otro      IN      A      72.52.4.120
```

4. Si el registro ya existe, se hacen los cambios solicitados.

Se guardan los cambios y se ejecuta el comando `svn diff` para revisar los cambios realizados.

Se ejecuta el comando `svn commit -m "cambios y razón"` para guardar los cambios en el repositorio, ya que no es necesario que pase por revisión.

Hecho esto, se comenta en el ticket informando al usuario que los nuevos registros DNS han sido añadidos o los ya existentes se han actualizados, y por último, se cierra el ticket o se responde el correo electrónico.

Como información adicional, una vez transcurrido el tiempo de la variable TTL se puede ejecutar el siguiente comando para el ejemplo anterior:

```
dig -tns otro.ejemplo.com [+short]
```

Este comando revisa si ya está propagado el registro DNS de manera externa, significando que cualquiera puede acceder a ese subdominio y será dirigido según la regla del mismo.

6.8 Añadir y actualizar paquetes en VMs

Dentro de este proceso, que inicia mediante la creación de un ticket en Jira, hay algunas variantes, es posible que solo se requiera instalar un paquete nuevo sin importar la versión o que se requiera una versión en específico.

Independientemente de esto, los paquetes deben agregarse a un archivo que es leído por un servicio que se ejecuta al iniciar una VM, estos archivos son nombrados como el nombre que tiene la VM.

Para instalar un paquete, dentro del archivo, existe una variable llamada `INSTALL_RPMS`, cada paquete que necesite ser instalado debe estar separado por espacio. Por ejemplo, si no importa la versión del paquete solo se pone el nombre del mismo separado por espacio; pero si se desea una versión en específico debe de agregarse el nombre del paquete seguido del signo igual y la versión del mismo sin dejar espacios entre sí. En caso de que se necesite actualizar o utilizar una versión más vieja del paquete solo es necesario cambiar la versión del mismo por la solicitada.

Una vez hecho esto, se ejecuta el comando `svn commit -m "cambios y razón"` para guardar los cambios en el repositorio, para esto tampoco es necesario que pase por revisión.

Los paquetes son extraídos de un repositorio interno, esto es por seguridad para que no pueda ser instalado cualquier paquete. Si el paquete que solicitan no está incluido en el repositorio será necesario agregarlo, solo es necesario buscar el archivo `.rpm` en internet compatible con CentOS 7 (en una página confiable u oficial) y añadirlo.

Una vez que estamos seguros que el paquete está en el repositorio, y ya está hecho el cambio en `svn` procedemos a acceder a la VM mediante `ssh`. Una vez dentro, ejecutamos `sudo service customize restart` para que se reinicie el servicio e instale el paquete que añadimos.

Para comprobar que el paquete está instalado podemos ejecutar `rpm -qa | grep "nombre_del_paquete"`, si esto nos muestra el nombre del paquete junto con la versión significa que ya hemos terminado.

Para finalizar, hacemos saber al usuario que ya se ha instalado y pueden hacer uso del mismo, y podemos cerrar el ticket.

6.9 Hosts en Nagios

Dentro de la empresa existen tres ambientes: staging, interno y externo.

En staging se encuentran los servidores de prueba o testing, en el interno los servidores de producción, que son donde existen los sitios web y en el externo los sitios web como se ven desde el mundo exterior.

Una vez explicado esto, también existen tres servidores de Nagios en la empresa, cada uno para cada ambiente de trabajo.

- En Nagios de staging se monitorean los servidores de prueba, ya sean sus servicios, el uso del CPU, espacio de almacenamiento, que estén activos y las páginas web que proveen.
- En Nagios interno, es igual que Nagios de staging pero para producción, se monitorean los servidores de producción, así como sus servicios, el uso del CPU, espacio de almacenamiento, que estén activos y las páginas web internas que proveen.
- En Nagios externo se monitorean las páginas web que proveen los servidores, pero vistas como se acceden en el exterior, así como sus certificados si es que tienen instalado alguno.

Para añadir un nuevo host a Nagios o modificar uno ya existente, debido a algún cambio en el servidor o la página web es necesario entrar a la carpeta de configuración de Nagios en svn, de ahí pasamos a escoger el ambiente que deseamos modificar y seleccionamos el archivo de hosts. Hacemos los cambios

necesarios y ejecutamos el comando `svn commit -m "cambios y razón"` ya que no es necesario que sea aprobado el cambio.

De igual manera si se requiere eliminar un host en Nagios, solo borramos la definición completa del host en el archivo de configuración y guardamos los cambios en svn.

Todos los cambios hechos en Nagios pueden tardar unos minutos en verse reflejados, esto es en lo que Nagios actualiza su configuración leyéndola de svn.

Cabe recalcar que para esta tarea en específico no entramos en mucho detalle ya que las posibilidades y variantes son muchas.

7 RESULTADOS OBTENIDOS

Los resultados de mis actividades dentro de la empresa son varios y dependen de cada actividad que realicé. Por ejemplo:

- Al crear una cuenta para un usuario y darle acceso a un servidor, se logra que el usuario pueda trabajar y/o resolver algún problema en el mismo.
- Al crear una cuenta en GitLab o svn el usuario puede tener acceso al código de su grupo de trabajo y agregar, modificar o eliminar el mismo.
- Al añadir un registro en el DNS logramos que una nueva página web sea dirigida al servidor o servidores que tienen el contenido de la misma.
- Al añadir un paquete a un servidor o servidores, logramos que los desarrolladores puedan disponer de las herramientas de trabajo que necesitan.
- Al añadir, modificar o eliminar hosts en Nagios, logramos tener una mejor visión de lo que está sucediendo en tiempo real en los servidores o en las páginas web para resolver los problemas que se puedan presentar.

Además de todo esto, los grandes conocimientos que he adquirido durante el tiempo que realicé mis prácticas profesionales para seguir desarrollándome profesionalmente en el área.

8 CONCLUSIONES Y RECOMENDACIONES

A mi parecer, las áreas que más deberían atenderse en el Departamento de Ingeniería Industrial, con respecto a las prácticas profesionales son:

- Seguir fomentando la participación de los alumnos en los proyectos de prácticas profesionales.
- Deberían impartirse pláticas o conferencias acerca de las experiencias de ex alumnos de la carrera.

Más que un proyecto, mis prácticas profesionales fueron una experiencia laboral, ya que me desarrollé en el área de trabajo, con un grupo de personas relativamente pequeño, en las actividades que se realizan día a día. Esto me llevo a adquirir una experiencia real en el área de administración de sistemas y servidores.

Realizar mis prácticas profesionales en una empresa internacional fue la experiencia más enriquecedora de mis estudios universitarios, por la enorme cantidad de tecnologías que se utilizan y que se pueden poner en práctica, además de todo lo que se puede aprender al mismo tiempo.

9 REFERENCIAS

<https://www.internetbrands.com/>

<https://www.linuxtrainingacademy.com/what-is-linux/>

<https://www.gnu.org/software/bash/>

<https://ryanstutorials.net/bash-scripting-tutorial/bash-script.php>

<https://www.vim.org/>

<https://subversion.apache.org/>

[https://en.wikipedia.org/wiki/Puppet_\(software\)](https://en.wikipedia.org/wiki/Puppet_(software))

<https://www.makeuseof.com/tag/virtual-machine-makeuseof-explains/>

<https://www.vmware.com/topics/glossary/content/hypervisor>

<https://about.gitlab.com/2015/05/18/simple-words-for-a-gitlab-newbie/>

<https://www.nagios.org/>

[https://en.wikipedia.org/wiki/Jira_\(software\)](https://en.wikipedia.org/wiki/Jira_(software))

<https://get.slack.help/hc/en-us/articles/115004071768-What-is-Slack->

<https://techterms.com/definition/ip>

<https://www.cloudflare.com/learning/dns/what-is-dns/>

<https://www.ssh.com/ssh/>

<https://www.ssh.com/ssh/key/>

https://docs.oracle.com/cd/E26180_01/Platform.94/ATGPersProgGuide/html/s0506_passwordhashing01.html



UNIVERSIDAD DE SONORA

COORDINACIÓN DIVISIONAL DE INGENIERIA

PRÁCTICAS PROFESIONALES

FPP-4

DEPARTAMENTO: Ingeniería Industrial y de Sistemas

UNIDAD REGIONAL CENTRO CAMPUS HERMOSILLO

REPORTE FINAL DE ACTIVIDADES

Periodo: Del 09 / julio / 2018 al 17 / agosto / 2018

Cantidad de 402 Horas de un total de 402 Avance: 100 %

Nombre del practicante: José Ismael Silvas Rodríguez

Expediente: 213214989 Programa Educativo (Licenciatura): IS1

Nombre del Programa/Proyecto: Control y monitoreo de usuarios y servidores

Datos de la Unidad Receptora (Razón Social): Vangtel Mexico S.A. de C.V.

Responsable de la Unidad Receptora (Nombre/Puesto): Mara Amaya/ Administrador de RH

Contacto: Teléfono/UR: 212 24 43 Ext. _____ Celular: _____

DESCRIPCIÓN GENERAL DE ACTIVIDADES

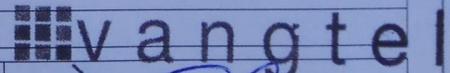
- Control de usuarios en Puppet. Agregar nuevas usuarios, modificar su información y eliminar cuentas, así como brindar acceso a los servidores con los permisos necesarios.
- Creación y eliminación de ventanitas en el repositorio SVN.
- Control de usuarios en GitLab: Agregar nuevos usuarios, modificar accesos a los grupos de trabajo y bloquear las cuentas requeridas.
- Añadir y actualizar paquetes en los servidores.
- Añadir, modificar y eliminar registros DNS.
- Añadir, modificar y eliminar cronjobs en los servidores.
- Bloquear IPs que atacan a los servidores.

RETROALIMENTACIÓN (Comentarios del tutor)

En caso de requerirse, anexar reportes, formatos, diagramas que apoyen las actividades realizadas.

Para las Ingenierías deberá anexar **reporte técnico** en archivo electrónico ≤ 2 MB y carta de terminación de prácticas firmada por el responsable de la empresa.

Observaciones Generales:



<p>Ismael S. <u>José Ismael Silvas Rodríguez</u> Nombre y firma del alumno</p>	<p><u>Federico M. Girett Galvan</u> Nombre y firma del tutor de prácticas profesionales UniSon.</p>	<p><u>Mara Amaya Torres</u> Vangtel Mexico S.A. de C.V. Nombre y firma del responsable de la Unidad Receptora Sello de la UR</p>
--	---	--

Original entregar en físico al Coordinador o Responsable de Prácticas Profesionales de la carrera.

Copia para Tutor de Prácticas Profesionales y Copia alumno.

Enviar en PDF los documentos al coordinador/responsable de prácticas profesionales de la carrera.

(25/04/2018)

Hermosillo, Son. Al 18 de septiembre 2018

DR. MARIO BARCELO VALENZUELA
PRESENTE

Por medio de la presente se hace constar que el C. JOSE ISMAEL SILVAS RODRIGUEZ, con expediente 213214989 de la carrera Ingeniería Sistemas de Información culminó su periodo de prácticas profesionales del 17 de mayo del 2018 al 17 de agosto de 2018, cumpliendo con un total de 402 horas dentro de nuestra empresa Vangtel México, S.A. de C.V. en el área de Internet Brands (NOC/Monitoreo de redes).

A petición del interesado y para los fines que se estimen convenientes, se extiende la presente quedando a sus órdenes.

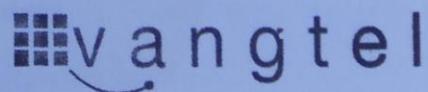
A handwritten signature in blue ink, appearing to be 'Mara', written over a horizontal line.

Mara Fernanda Amaya Cárdenas
Operaciones RECURSOS HUMANOS

www.intugo.co
VANGTEL MÉXICO S.A. DE C.V.
Tel. +52 662 212 2443

Bld. Río Sonora #110 Int.13,
Col. Hacienda de la Flor,
C.P. 83090, Hermosillo, Sonora, MX.

 /trabajaenintugo  /intugomx

The logo for 'vangtel' features the word in a lowercase, sans-serif font. To the left of the text is a stylized icon consisting of a grid of small squares, with a curved line underneath it.

Vangtel México, S.A. de C.V.
Coordinación R.H.
VME050706UUA