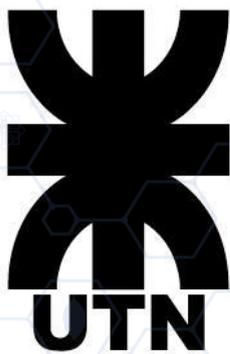


Practica Profesional



Ministerio de
SALUD
Gobierno de Entre Ríos

**Sistema Web para la Gestión y Análisis de Redes
Informáticas**

Alumno: FONTANA, Cristian Ezequiel

Periodo mayo / junio 2024

Resumen Ejecutivo

Durante mi pasantía en el Departamento de Informática del Ministerio de Salud, desarrollé una aplicación web destinada a mejorar la gestión y el monitoreo de la red informática. El proyecto incluyó la implementación de un sistema de registro de usuarios y la creación de reportes específicos para facilitar el trabajo diario de los técnicos en los **casi 700 dispositivos que se encuentran en el edificio**. Utilicé tecnologías como Angular, SpringBoot y Hibernate, enfrentando desafíos como la migración de bases de datos de PostgreSQL a MySQL y la optimización de tiempos de respuesta ante inundaciones ICMP. La aplicación resultante mejora la eficiencia operativa, proporcionando monitoreo en tiempo real y una gestión optimizada de recursos de red.

ÍNDICE

Introducción.....	5
I. Aspectos Generales.....	6
1.1 Antecedentes.....	6
1.2 Objetivos.....	6
1.2.1 General.....	6
1.2.2 Específico.....	8
1.3 Delimitación de la practica profesional.....	10
1.3.1 Objetivos específicos:.....	10
1.3.2. Ámbito de trabajo:.....	10
1.3.3. Tecnología específica:.....	10
1.3.4. Base de datos:.....	11
1.4 Limitaciones.....	11
II. Evaluación Institucional.....	12
2.1 Descripción General de la Institución.....	12
2.2 Objetivos de la Institución.....	13
2.3 Visión.....	13
2.4 Misión.....	14
2.5 Actividades.....	14
2.6. Estructura organizacional.....	16
2.7 Descripción del grupo humano de trabajo.....	17
2.8 Escenario de trabajo.....	18
Escenario Físico.....	18
Ambiente de Trabajo.....	18
Tecnologías y Herramientas.....	19
Estructura del Equipo.....	19
Procesos y Metodologías.....	20
III Desarrollo de la Practica Profesional.....	20
3.1 Descripción de actividades desarrolladas en la práctica Profesional.....	20
Stack Tecnológico Utilizado.....	20

Presentación del Frontend de la Aplicación.....	23
Funcionalidades Principales:.....	23
Diseño y Usabilidad:.....	24
Accesibilidad para Todos los Usuarios:.....	24
Futuros desarrollos:.....	25
Funcionalidad de Búsqueda por IP.....	25
Explicación del Código y Funcionalidades.....	29
Funcionamiento del Código.....	29
Funcionalidad.....	31
Funcionalidad de Búsqueda por Rango IP.....	31
Objetivo:.....	32
Detalles de la Implementación:.....	33
Explicación del Código y Funcionalidades de Búsqueda por Rango de IP.....	35
Funcionalidades del Código:.....	35
Funcionalidades:.....	38
Funcionalidad de Búsqueda por Oficinas.....	40
Detalles de Implementación.....	41
Funcionalidad ABM Usuarios.....	46
Objetivo.....	47
Modo de Uso.....	47
Detalles de Implementación.....	49
Funcionalidad de ABM Oficinas.....	51
Objetivo.....	51
Modo de Uso.....	52
Detalles de Implementación.....	53
Explicación del Código y Funcionalidades.....	54
Conclusiones.....	55
Aprendizajes obtenidos en la realización de la práctica.....	55
Comentarios personales del trabajo realizado.....	56
Conclusiones Generales.....	57

Introducción

La presente pasantía se desarrolló en el Departamento de Informática del Ministerio de Salud, con el objetivo de crear una aplicación web que facilite la gestión y el monitoreo de la red informática de la institución. La necesidad de este proyecto surgió debido a la falta de herramientas eficientes para la administración de nodos y la monitorización en tiempo real, lo cual dificulta el trabajo diario de los técnicos e ingenieros encargados del mantenimiento de la red.

Para abordar esta problemática, se desarrolló un sistema utilizando tecnologías modernas como Angular para el frontend, SpringBoot para el backend y Hibernate para la gestión de bases de datos. Este sistema incluye funcionalidades para el registro y gestión de usuarios, así como la generación de reportes detallados sobre el estado de la red y los dispositivos conectados. A lo largo de la pasantía, se enfrentaron varios desafíos técnicos, incluyendo la migración de bases de datos de PostgreSQL a MySQL y la optimización del manejo de inundaciones ICMP.

En este informe se detalla todo el proceso de desarrollo de la aplicación, los métodos utilizados, los desafíos enfrentados y los resultados obtenidos, proporcionando una visión generalizada del trabajo realizado durante la pasantía y su impacto en la mejora de la gestión de la red informática del Ministerio de Salud.

I. Aspectos Generales

1.1 Antecedentes

En el Departamento de Informática del Ministerio, se llevan a cabo, entre otras actividades, tareas de mantenimiento de computadoras. Inicialmente, cuando surgía un problema técnico con una computadora dentro del edificio, se recibía una llamada telefónica detallando el problema. Se solicitaba el nombre asignado a la PC, siguiendo el formato "MASTERXXX" (donde XXX es un número incremental único). Con esta información, el personal accedía al sistema Active Directory para obtener la IP asignada y luego ejecutaba un ping desde la consola para verificar la conectividad de la PC, procediendo con el mantenimiento necesario según los resultados obtenidos. (ver fotografías en anexo)

1.2 Objetivos

1.2.1 General

El objetivo principal de la pasantía es la creación de un sistema web con un diseño de estilo dashboard iconizado, que sea de fácil acceso y uso. Este sistema debe contener una serie de herramientas que faciliten el trabajo diario de los técnicos e ingenieros responsables del mantenimiento de las diversas redes informáticas dentro del edificio.

El sistema está diseñado para ofrecer una interfaz intuitiva y amigable que permita a los usuarios realizar tareas críticas con mayor eficiencia. Las herramientas integradas en el dashboard incluyen funcionalidades para la gestión de usuarios, la administración de oficinas y la monitorización en tiempo real de dispositivos. Cada herramienta está optimizada para simplificar procesos que antes eran tediosos y propensos a errores, como la gestión manual de datos en hojas de cálculo (excel).

Además, el sistema debe proporcionar capacidades avanzadas de generación de reportes, permitiendo a los usuarios obtener información detallada sobre el estado de la red, la conectividad de los dispositivos y la organización interna de los recursos tecnológicos. Estos reportes son esenciales para la toma de decisiones informadas y la resolución rápida de problemas técnicos.

En resumen, la creación de este sistema web busca mejorar significativamente la operatividad y la gestión de las redes informáticas, proporcionando a los técnicos e ingenieros del Ministerio una herramienta poderosa y eficiente para sus labores diarias.

1.2.2 Específico

Desarrollo del Sistema de Registro de Usuarios: En primer lugar, se solicitó el desarrollo de un sistema de registro de usuarios que permitiera el alta, baja y modificación de usuarios existentes. Este sistema debía reemplazar el uso de una planilla Excel que generaba un trámite engorroso al realizar actualizaciones y mantenimientos. La implementación de este sistema buscaba mejorar la eficiencia en la gestión de usuarios, reducir errores y facilitar el proceso de mantenimiento.

Desarrollo de Reportes: En segundo lugar, se me encargó la tarea de desarrollar una serie de reportes que permitieran organizar mejor la información y optimizar la gestión interna. Estos reportes incluyen funcionalidades como la visualización del estado de conexión de dispositivos en tiempo real y la organización de información específica por oficinas y rangos de IP. Estas herramientas proporcionarán una visión clara y precisa de la infraestructura de red, a continuación detallaremos dos de los reportes.

- **Reporte por oficina:** Este informe proporcionará una visualización práctica de una lista detallada de todos los dispositivos que se encuentran en una oficina específica. Este informe detallado mostrará los nombres y el estado de conexión en tiempo real de cada dispositivo utilizando el protocolo ICMP (Internet Control Message Protocol). Estos datos incluirán

información como si los dispositivos están conectados o desconectados.

Este tipo de reporte es crucial para la gestión eficiente de los recursos en cada oficina porque facilita la identificación y resolución rápida de problemas de conectividad. Además, mejora la gestión y el mantenimiento de los equipos del edificio al organizar la infraestructura tecnológica de manera clara y concisa.

- **Reporte por Rango IP:** El reporte por rango IP permitirá ver el estado de conexión de la PC de un rango de direcciones IP en tiempo real. Este informe proporcionará información detallada sobre cada dispositivo dentro del **rango**, incluido el estado de conexión (conectado o desconectado) y el nombre del PC, conocido como nombre master (formato "MASTERXXX"). El reporte también proporcionará una visión organizada y clara del rendimiento y la disponibilidad de los dispositivos, lo que facilitará la identificación de problemas y la gestión eficiente de la red. Este informe es esencial para el monitoreo continuo y la resolución rápida de problemas de infraestructura.

1.3 Delimitación de la práctica profesional

1.3.1 Objetivos específicos:

La pasantía se centró en el desarrollo de una aplicación web con un diseño de estilo dashboard iconizado, accesible y fácil de usar, que centralice las funcionalidades necesarias para la gestión y monitoreo de la red informática.

1.3.2. Ámbito de trabajo:

La pasantía se delimitó al Departamento de Informática, más precisamente en la División Desarrollo y no incluyó las visitas a otros departamentos del edificio.

1.3.3. Tecnología específica:

Se utilizó un stack tecnológico específico como Angular en Frontend y Java con SpringBoot en Backend, compartiendo conocimientos con otro programador que también utilizaba las mismas herramientas.

1.3.4. Base de datos:

En el proyecto se optó por cambiar la base de datos de PostgreSQL a MySQL debido a cuestiones de compatibilidad con el motor de base de datos existente en el Ministerio. Esta decisión fue importante para asegurar la integración dentro del entorno tecnológico actual del Ministerio de Salud.

1.4 Limitaciones

Una de las limitaciones encontradas durante la pasantía fue la falta de acceso a los datos de los usuarios. Esta restricción se debió a la confidencialidad de los mismos, lo cual es comprensible y necesario para proteger la privacidad y la seguridad de los usuarios.

Aunque esta limitación impide un acceso directo a los datos reales, se adoptaron medidas para simular la gestión de usuarios en un entorno de prueba, lo que permitió continuar con el desarrollo y la implementación del sistema sin comprometer la integridad de los datos mencionados anteriormente.

II. Evaluación Institucional

2.1 Descripción General de la Institución

El Ministerio de Salud de Entre Ríos es la autoridad sanitaria provincial encargada de planificar, coordinar y supervisar los servicios de salud en toda la provincia. Su estructura organizativa incluye diversas áreas y departamentos especializados, tales como la vigilancia epidemiológica, la cual tomó gran relevancia con la pandemia covid 19, la atención primaria, las residencias médicas, bioingeniería, entre otros. Es una entidad comprometida con la salud de la comunidad, desarrollando políticas y estrategias que promuevan el bienestar y prevengan enfermedades a lo largo y ancho de la provincia.

El mismo se logra a través de la gestión de múltiples programas nacionales y provinciales que buscan mejorar la salud pública como los programas de prevención de VIH, Diabetes y la prevención de enfermedades crónicas entre otras tantas. También proporciona formación y recursos a los profesionales de la salud para garantizar una atención sanitaria óptima para todos los entrerrianos

2.2 Objetivos de la Institución

El objetivo del Ministerio de Salud de la provincia es asistir al Poder Ejecutivo en todos los asuntos relacionados con la salud de la población y la promoción de conductas saludables. Entre sus funciones, se destacan la ejecución de planes y programas conforme a directivas provinciales y nacionales, la fiscalización de servicios y establecimientos de salud, y la elaboración de normas que regulan la práctica médica y odontológica, garantizando la accesibilidad y calidad en la atención para los entrerrianos. Además, regula, adjudica y controla los productos y equipos de salud que se adquiere para los Hospitales y Centros de salud, además coordina el sistema de salud con los municipios y difunde información para reducir desigualdad de acceso a la salud.

2.3 Vision

El Ministerio de Salud de Entre Ríos planea ser un modelo de referencia a nivel nacional en el fomento de la salud y el bienestar de nuestros ciudadanos. En el futuro, se espera incorporar más equipos médicos de alta tecnología para mejorar la calidad y eficiencia de los servicios de salud. Además, se asegurará de que ningún entrerriano quede fuera del sistema de salud, garantizando un acceso equitativo y avanzado a la atención médica para todos los habitantes de la provincia.

2.4 Misión

El Ministerio de Salud de Entre Ríos es la institución que orienta y coordina a los diferentes actores sociales en la ejecución de acciones para proteger y promover la salud física, mental y social de la población. En coordinación con el Ministerio de Salud de la Nación, implementa programas de prevención y promoción de la salud, campañas de vacunación, y acciones conjuntas para la vigilancia epidemiológica. Además, trabaja en el desarrollo de políticas sanitarias, la capacitación de profesionales de la salud y la implementación de estrategias para mejorar la infraestructura y el acceso a servicios de salud.

2.5 Actividades

El Ministerio de Salud de Entre Ríos lleva a cabo una amplia gama de actividades para mejorar la salud pública y garantizar un sistema sanitario equitativo y eficiente. Entre las principales actividades se encuentran:

1. **Prevención y promoción de la salud:** El Ministerio realiza campañas de vacunación masiva, prevención de enfermedades respiratorias, dengue, VIH/SIDA, y promueve hábitos de vida saludable. También lleva a cabo campañas de concientización sobre la importancia de la salud sexual y reproductiva.

2. **Innovación tecnológica:** Se enfoca en la implementación de tecnologías modernas en el sistema de salud, como la adopción de recetas electrónicas, historia clínica digital y telemedicina. Estas innovaciones buscan mejorar la eficiencia y accesibilidad del sistema de salud, facilitando el acceso a servicios médicos de calidad a todos los ciudadanos.

3. **Formación y capacitación:** El Ministerio desarrolla programas de formación y capacitación para profesionales de la salud, asegurando que el personal sanitario esté continuamente actualizado con las últimas prácticas y conocimientos en medicina y salud pública. Ofrece becas, cursos y programas de residencias médicas para fortalecer las competencias profesionales.

4. **Vigilancia epidemiológica:** Se dedica a la vigilancia y control de brotes epidémicos y enfermedades infecciosas, monitoreando la situación sanitaria en todo el país. Implementa sistemas de alerta temprana y protocolos de respuesta rápida para mitigar el impacto de posibles epidemias y pandemias.

5. Coordinación interinstitucional Colabora estrechamente con otras entidades, tanto nacionales como provinciales, así como con organizaciones internacionales, para mejorar la atención sanitaria. Esta coordinación permite una respuesta más eficiente a los desafíos de salud pública de hoy día, facilitando el intercambio de información y recursos.

6. Gestión y desarrollo institucional: El Ministerio trabaja en la mejora de la infraestructura sanitaria, incluyendo la construcción y renovación de hospitales y centros de salud de Paraná y del interior, así como en la dotación de equipamiento médico avanzado. También se enfoca en la optimización de los procesos administrativos y la transparencia en la gestión de recursos.

2.6. Estructura organizacional

El presente organigrama del Ministerio de Salud de Entre Ríos refleja la estructura organizativa y las relaciones funcionales dentro de la institución, recientemente reorganizada bajo el nuevo gobierno. Este organigrama destaca las principales áreas de gestión y operativas, desde el Ministro de Salud hasta las divisiones específicas, como la Dirección General de Recursos Materiales Hospitalarios y el Departamento de Informática.

Este último es importante, dado que fue el ámbito de mi pasantía.



2.7 Descripción del grupo humano de trabajo

Durante mi pasantía, trabajé con un grupo humano altamente colaborativo y siempre dispuesto a ayudar. Mostrando los sistemas que han desarrollado, los que están trabajando y las experiencias personales. Este equipo, compuesto por profesionales de diversas áreas, Licenciados en Sistemas, Ingenieros Electrónicos y Técnicos en informática. Desde el

inicio, mostraron una actitud abierta y acogedora, compartiendo sus conocimientos y experiencias en la plataforma de desarrollos Angular, lo cual me sirvió para resolver varios inconvenientes que me encontré en el desarrollo. La comunicación fluida y el apoyo fueron clave para crear un ambiente de trabajo interesante. Su disposición a asistir y guiarme fue fundamental para mi desarrollo profesional y personal durante la pasantía.

2.8 Escenario de trabajo

Escenario Físico

El entorno de trabajo se desarrolló principalmente de forma presencial, utilizando mi notebook personal para todas las tareas de desarrollo. Sin embargo, la empresa cuenta con oficinas bien equipadas, con áreas de trabajo abiertas y salas de reuniones para facilitar la colaboración.

Ambiente de Trabajo

El Departamento Informática se caracteriza por un ambiente profesional y serio . Cada jefe de área es responsable de sus respectivos equipos, lo que facilita una gestión clara y

eficiente. Mi tutor durante la pasantía fue el Lic. Dario Reisenauer, quien proporcionó una guía importante y valiosa en todo el trayecto del desarrollo del sistema de análisis.

Tecnologías y Herramientas

En la división de Desarrollo, utilizamos diversas herramientas y tecnologías para llevar a cabo nuestras tareas, con un enfoque especial en plataformas de desarrollo de software y gestión de proyectos. Mi trabajo principal se realizaba en mi notebook personal, lo cual me proporcionó flexibilidad y autonomía.

Estructura del Equipo

El Departamento se divide en tres áreas o divisiones principales: Desarrollo, Ingeniería en Redes y Mantenimiento. Trabajé en la división de Desarrollo, donde cada miembro del equipo tenía roles y responsabilidades claramente definidos. La estructura jerárquica permitía una comunicación fluida.

Procesos y Metodologías

Se seguían métodos con consultas constantes para coordinar el progreso y resolver cualquier inconveniente. El escenario de trabajo durante mi pasantía fue muy agradable, con un ambiente de apoyo y colaboración que me permitió aprender y desarrollarme profesionalmente.

III Desarrollo de la Practica Profesional

3.1 Descripción de actividades desarrolladas en la práctica Profesional

Stack Tecnológico Utilizado

Para el desarrollo de la aplicación web se utilizó un stack tecnológico moderno que garantiza una gran compatibilidad y facilidad de mantenimiento.

Frontend:

- *Angular*: Framework principal para la construcción de la interfaz de usuario. Angular permite la creación de aplicaciones web dinámicas y responsive (adaptables).

- *Bootstrap*: Utilizado para el diseño y la disposición de los componentes visuales, asegurando una interfaz limpia y moderna.

- *FontAwesome*: Proporciona una amplia variedad de iconos que mejoran la estética del sistema.

Backend:

- *Java con SpringBoot*: Framework utilizado para el desarrollo del backend, proporcionando un entorno robusto y escalable para la implementación de las API's y la lógica del servidor.

- *Maven*: Herramienta de gestión de dependencias y construcción de proyectos, facilitando la administración de bibliotecas y plugins necesarios para el desarrollo.

Base de Datos:

- *PostgreSQL y MySQL*: Inicialmente se utilizó PostgreSQL ya que en la facultad utilizamos ampliamente esta base de datos, pero posteriormente se migró a MySQL debido

a requisitos de compatibilidad con el entorno existente, este tema se tratará más adelante en este informe.

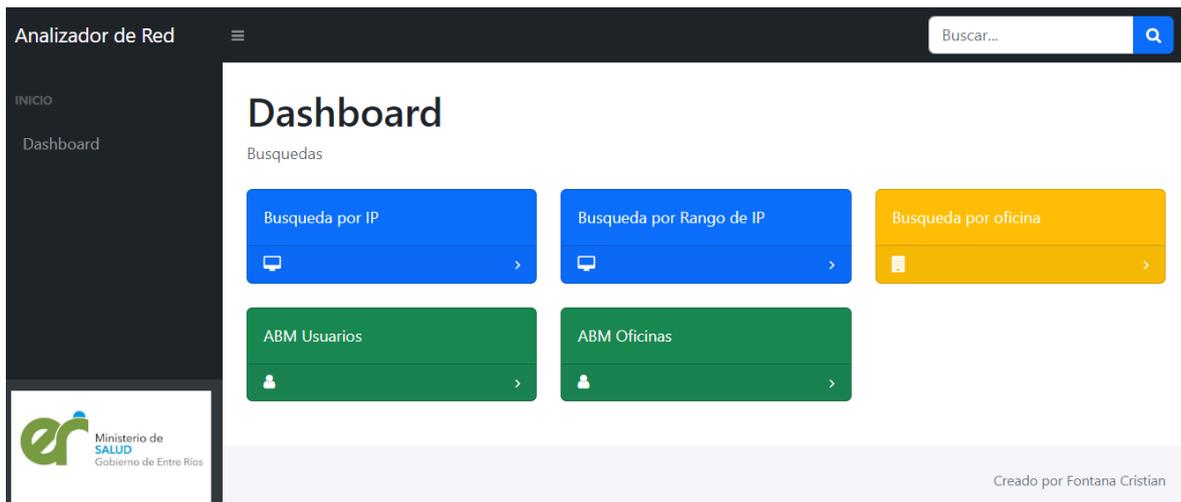
- *Hibernate*: ORM que facilita la interacción entre la aplicación Java y las bases de datos relacionales, permitiendo una gestión eficiente de las transacciones y consultas.

Comunicación y Monitoreo:

- *Protocolo ICMP*: Utilizado para monitorear la conectividad de los dispositivos en la red, permitiendo la detección y resolución rápida de problemas de conexión.

Este stack tecnológico fue coordinado con Dario para asegurar una integración eficiente, escalabilidad y un desarrollo ágil, garantizando que la aplicación pueda evolucionar y adaptarse a futuras necesidades y mejoras.

Presentación del Frontend de la Aplicación



La aplicación desarrollada cuenta con un frontend intuitivo y fácil de usar, diseñado para facilitar la gestión y monitoreo de la red informática del Ministerio de Salud de Entre Ríos.

El mismo proporciona acceso rápido a las principales funcionalidades del sistema.

Funcionalidades Principales:

- *Búsqueda por IP*: Permite realizar búsquedas detalladas de dispositivos mediante su dirección IP.

- *Búsqueda por Rango de IP*: Facilita la visualización del estado de conexión de un rango específico de IPs.

- *Búsqueda por Oficina:* Muestra un listado de dispositivos ubicados en una oficina específica.
- *ABM Usuarios:* Gestiona el alta, baja y modificación de usuarios del sistema.
- *ABM Oficinas:* Administra el alta, baja y modificación de oficinas.

Diseño y Usabilidad:

El diseño del dashboard es limpio y ordenado, utilizando tecnología Bootstrap para la disposición de componentes estilo “cards” y FontAwesome para los iconos, lo que asegura una experiencia de usuario agradable y eficiente. Las opciones están claramente segmentadas y accesibles, lo que reduce el tiempo necesario para navegar y utilizar las funcionalidades del sistema.

Accesibilidad para Todos los Usuarios:

Una característica destacada de esta aplicación es su diseño intuitivo, que permite su uso no solo por técnicos e ingenieros, sino también por personal administrativo. La interfaz

amigable asegura que cualquier usuario, independientemente de su nivel técnico, pueda utilizarla eficazmente.

Futuros desarrollos:

Para el diseño de la aplicación, se utilizó un **Navbar** y un **buscador** en la interfaz principal. Estos elementos no solo mejoran la navegación y la experiencia del usuario, sino que también permiten la posibilidad de añadir más funcionalidades al sistema en el futuro. Esto asegura que la aplicación pueda evolucionar y adaptarse a nuevas necesidades sin comprometer la usabilidad y la eficiencia.

Funcionalidad de Búsqueda por IP

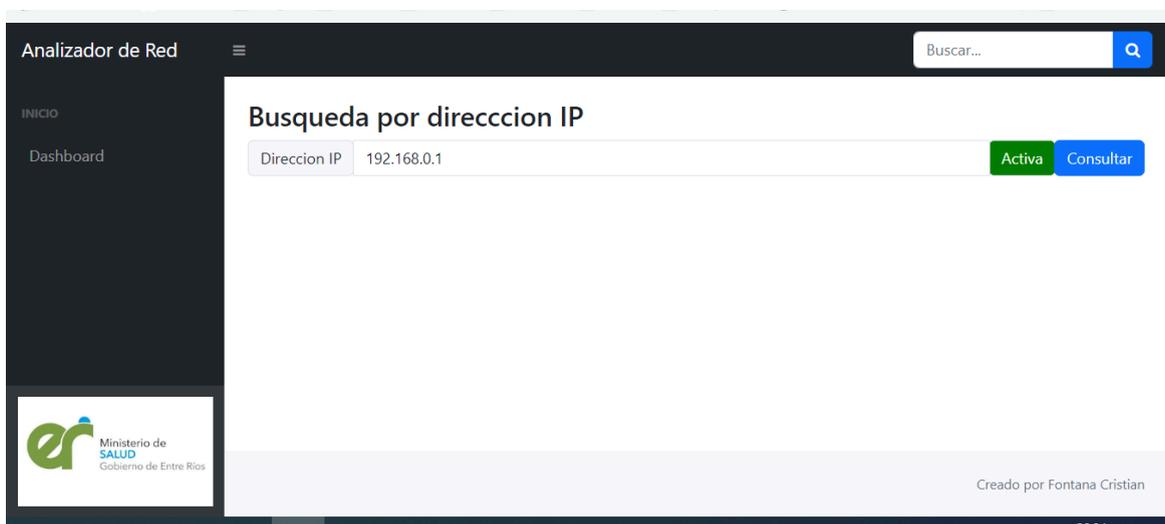
La funcionalidad de "Búsqueda por IP" fue la primera en desarrollarse para comenzar a

integrar y comprender los comandos del sistema operativo, como **cmd** y **ping**, y sus respectivas funcionalidades. Este componente permite verificar el estado de conectividad de una dirección IP específica en la red, utilizando comandos del sistema operativo para enviar paquetes de red, recibirlos y procesar sus respuestas. La implementación de esta



funcionalidad fue crucial para sentar las bases del sistema de monitoreo en tiempo real dentro de la aplicación.

. En la aplicación se permite a los usuarios verificar el estado de conexión de un dispositivo específico mediante su dirección IP.



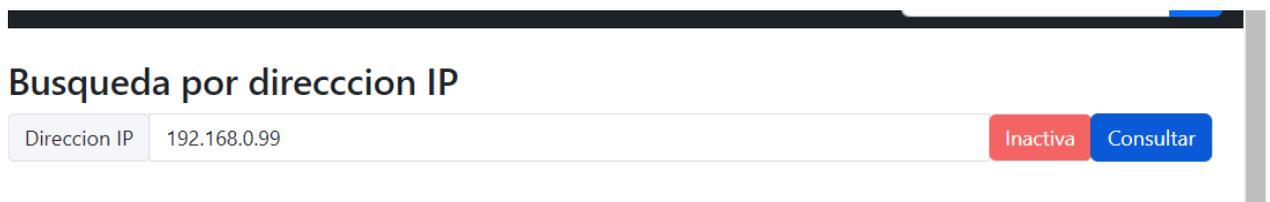
Objetivo:

El propósito principal es permitir una rápida identificación y diagnóstico de la conectividad de un dispositivo dentro de la red local.

Detalles de la Implementación:

- Parámetro de Búsqueda: Dirección IP.

- Presentación: La información se muestra en un formato claro que indica si la dirección IP está activa o no.

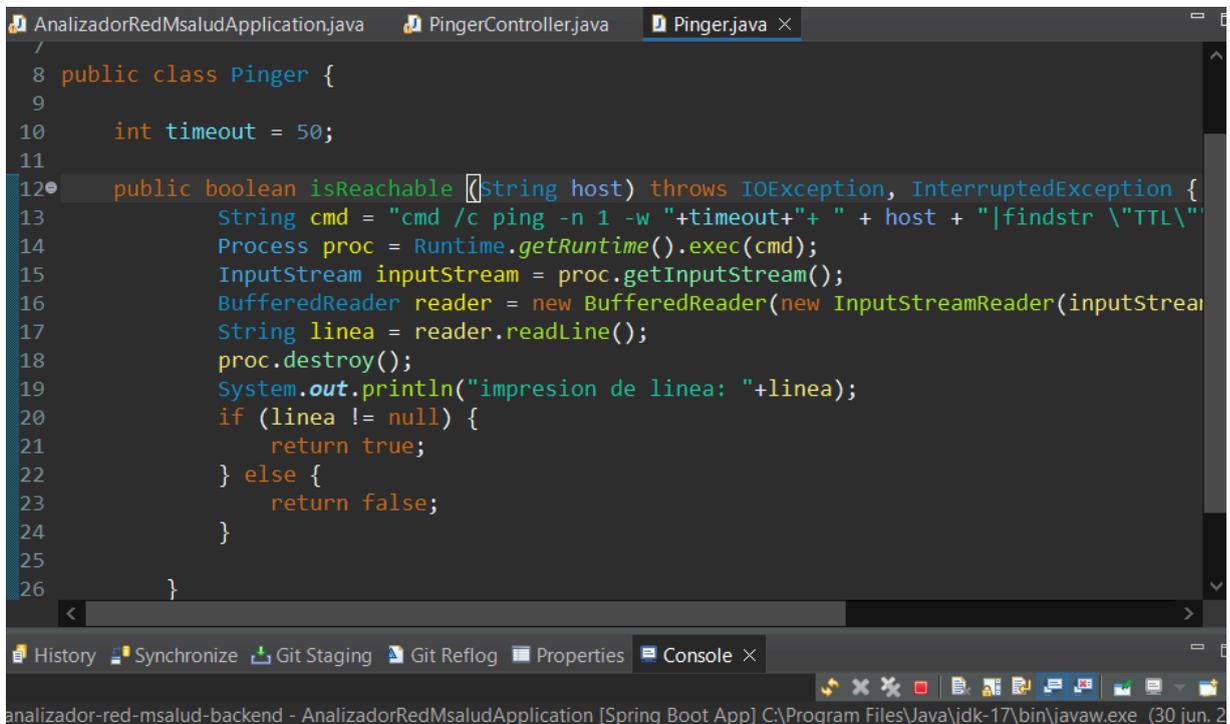


Busqueda por direccion IP

Direccion IP 192.168.0.99 Inactiva Consultar

Proceso de Búsqueda:

1. El usuario introduce la dirección IP en el campo correspondiente y hace clic en "Consultar".
2. La aplicación envía una solicitud al backend, donde se ejecuta un comando ping para la dirección IP proporcionada.
3. El backend utiliza la clase `Pinger` para realizar la verificación de conectividad:



```

8 public class Pinger {
9
10     int timeout = 50;
11
12     public boolean isReachable(String host) throws IOException, InterruptedException {
13         String cmd = "cmd /c ping -n 1 -w "+timeout+" " + host + "|findstr \"/>

```

4. Dependiendo del resultado, se muestra en la interfaz si la dirección IP está **activa (conectada) o no**.

Esta funcionalidad es esencial para el monitoreo, proporcionando una herramienta sencilla de uso rápido y consulta al al paso, la cual puede utilizar cualquier miembro del personal, técnico o administrativo.

Explicación del Código y Funcionalidades

El código proporcionado es una clase en Java llamada `Ping`, que contiene un método estático `isReachable`. Este método verifica si una dirección IP es alcanzable utilizando el comando `ping` del sistema operativo. Aquí está la explicación detallada:

Funcionamiento del Código

1. Comando Ping:

```
String cmd = "cmd /c ping -n 1 -w 50 " + host + "|findstr \"TTL\"";
```

Este comando ejecuta un ping a la dirección IP especificada (`host`), enviando solo un paquete (`-n 1`) y esperando hasta 50 milisegundos (`-w 50`). El resultado se filtra para buscar la cadena "TTL", que indica una respuesta exitosa.

2. Ejecución del Comando:

```
Process proc = Runtime.getRuntime().exec(cmd);
```

Se ejecuta el comando ping utilizando ``Runtime.getRuntime().exec(cmd)``.

3. Lectura de la Salida:

```
InputStream inputStream = proc.getInputStream();  
BufferedReader reader = new BufferedReader(new InputStreamReader(inputStream));  
String linea = reader.readLine();  
proc.destroy();
```

La salida del comando se lee usando un ``BufferedReader``. Si la salida contiene la cadena "TTL", el método ``isReachable`` devolverá ``true``, indicando que la dirección IP es alcanzable. Si no, devolverá ``false``.

4. Impresión de Resultado:

```
System.out.println("impresion de linea: " + linea);
```

Esta línea imprime la respuesta del ping para propósitos de depuración.

Funcionalidad

- Verificación de Conectividad: El método `isReachable` permite verificar si un host es alcanzable en la red.
- Uso de Comandos del Sistema: Utiliza comandos del sistema operativo para realizar la verificación, lo que hace que sea una solución eficiente y directa.
- Filtrado de Resultados: Filtra la salida del comando para buscar indicaciones específicas de éxito (cadena "TTL").

Funcionalidad de Búsqueda por Rango IP

La funcionalidad de "Búsqueda por Rango de IP" fue desarrollada para permitir la verificación masiva del estado de conectividad



de dispositivos dentro de un rango específico de direcciones IP. Esta herramienta es esencial para el monitoreo eficiente de grandes segmentos de la red, facilitando a los usuarios identificar rápidamente qué dispositivos están activos o inactivos.



The screenshot shows a web application titled 'Analizador de Red'. It features a search interface for IP ranges. The search results are displayed in a table with the following data:

Direccion IP	Nombre Master	Estado
192.168.0.1	master1000	Activa
192.168.0.2	master1001	Inactiva
192.168.0.3	master1002	Inactiva
192.168.0.4	master1004	Inactiva
192.168.0.5	N/D	Inactiva

Objetivo:

El objetivo es facilitar la verificación masiva del estado de dispositivos en un rango de direcciones IP, mostrando su conectividad y el nombre asignado a cada dispositivo si esta previamente cargado en la base de datos de este sistema.

Detalles de la Implementación:

- Parámetros de Búsqueda: IP inicial y IP final del rango a verificar.
- Presentación: Una tabla que muestra la dirección IP, el nombre master y el estado de conectividad (activa o inactiva).

```

AnalizadorRedMsaludApplication.java | PingerController.java | Pinger.java
54
55 @GetMapping(value = "/api/pinger-rango")
56 public List<PingResult> getPingerRango(@RequestParam String ipInicio, @RequestParam String ipFin) {
57
58
59     List<PingResult> results = new ArrayList<>();
60     // List<PingResult> results = pingRange(ipInicio,ipFin, timeout);
61
62     long start = ipToLong(ipInicio);
63     long end = ipToLong(ipFin);
64
65     Pinger pinger = new Pinger();
66
67     for (long i = start; i <= end; i++) {
68         String ip = LongToIp(i);
69         try {
70             InetAddress inet = InetAddress.getByName(ip);
71             //boolean reachable = inet.isReachable(timeout);
72             boolean reachable = pinger.isReachable(ip);
73             String nombre_master = usuarioService.getNombreMasterByIp(ip);
74             String estado = reachable ? "Activa" : "Inactiva";
75             results.add(new PingResult(ip, estado, nombre_master));
76         } catch (Exception e) {
77             results.add(new PingResult(ip, "Inactiva", "N/D"));
78             e.printStackTrace();
79         }
80     }
81     return results;
82 }
83
84
85
86 public static long ipToLong(String ipAddress) {
87     String[] ipParts = ipAddress.split("\\.");
88     long ip = 0;
89     for (int i = 0; i < 4; i++) {
90         ip <<= 8;
91         ip |= Integer.parseInt(ipParts[i]);
92     }
93     return ip;
94 }
95
96 public static String longToIp(long ip) {
97     return ((ip >> 24) & 0xFF) + "." +
98         ((ip >> 16) & 0xFF) + "." +
99         ((ip >> 8) & 0xFF) + "." +
100        (ip & 0xFF);

```

4. El resultado muestra el estado de cada IP, indicando si está activa o inactiva, junto con el nombre master del dispositivo.

Explicación del Código y Funcionalidades de Búsqueda por Rango de IP

El código se encarga de verificar la conectividad de un rango de direcciones IP y recopilar resultados sobre su estado y nombre asociado. Aquí se desglosa su funcionamiento:

Funcionalidades del Código:

1. Recepción de Parámetros:

```
@GetMapping(value = "/api/pinger-rango") public List<PingResult>  
getPingerRango(@RequestParam String ipInicio, @RequestParam String ipFin) {
```

El método `getPingerRango` recibe los parámetros `ipInicio` e `ipFin`, que determinan el rango de IPs a verificar.

2. Conversión de IP a Long:

```
long start = ipToLong(ipInicio);  
long end = ipToLong(ipFin);
```

Se convierten las direcciones IP de su formato de cadena a un formato numérico ('long') para facilitar la iteración.

3. Inicialización del Pinger:

```
Pinger pinger = new Pinger();
```

Se crea una instancia de la clase 'Pinger' para realizar las verificaciones de conectividad.

4. Iteración y Verificación:

```
for (long i = start; i <= end; i++) {  
    String ip = longToIp(i);  
    try {  
        boolean reachable = pinger.isReachable(ip);  
        String nombre_master = usuarioService.getNombreMasterByIp(ip);  
        String estado = reachable ? "Activa" : "Inactiva";  
        results.add(new PingResult(ip, estado, nombre_master));  
    } catch (Exception e) {  
        results.add(new PingResult(ip, "Inactiva", "N/D"));  
        e.printStackTrace();  
    }  
}
```

```
}
```

Se itera sobre el rango de IPs, convirtiendo cada valor numérico de vuelta a su formato de cadena y utilizando el método `isReachable` de `Pinger` para verificar la conectividad. Se obtiene el nombre asociado (`nombre_master`) y se determina el estado (`Activa` o `Inactiva`). Los resultados se almacenan en una lista de `PingResult`.

5. Conversión de IP de y hacia Long:

```
public static long ipToLong(String ipAddress) {  
    String[] ipParts = ipAddress.split("\\.");  
    long ip = 0;  
    for (int i = 0; i < 4; i++) {  
        ip <<= 8;  
        ip |= Integer.parseInt(ipParts[i]);  
    }  
    return ip;  
}
```

```
public static String longToIp(long ip) {  
    return ((ip >> 24) & 0xFF) + "." +
```

```
((ip >> 16) & 0xFF) + "." +  
((ip >> 8) & 0xFF) + "." +  
(ip & 0xFF);  
}
```

Estos métodos convierten una dirección IP en formato de cadena a un número `long` y viceversa, facilitando la iteración sobre el rango de direcciones.

6. Retorno de Resultados:

```
return results;
```

Finalmente, se devuelve la lista de resultados, que contiene la información sobre la conectividad de cada dirección IP en el rango especificado.

Funcionalidades:

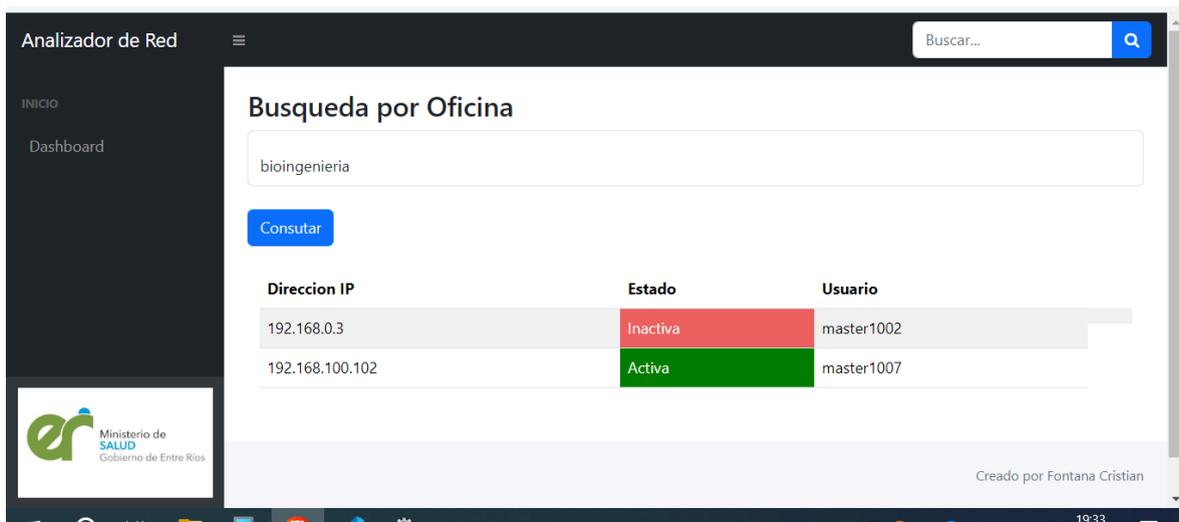
- Recepción de un rango de IPs: Permite al usuario especificar un rango de direcciones IP para verificar.

- Verificación de Conectividad: Utiliza el comando `ping` para comprobar si las direcciones IP son alcanzables.
- Recopilación de Información: Obtiene el estado de conectividad y el nombre asociado a cada dirección IP.
- Conversión de Formato: Convierte direcciones IP entre formatos de cadena y numérico para facilitar el procesamiento.
- Retorno de Resultados: Proporciona una lista de resultados detallando el estado de cada IP en el rango especificado.

Esta funcionalidad es fundamental para el monitoreo eficiente de grandes segmentos de la red, permitiendo a los usuarios obtener una visión rápida y precisa del estado de conectividad de múltiples dispositivos.

Funcionalidad de Búsqueda por Oficinas

La funcionalidad de "Búsqueda por Oficina" fue diseñada para permitir la verificación del estado de conectividad de los dispositivos ubicados en una oficina específica. Esta herramienta facilita la gestión y el monitoreo centralizado de los recursos tecnológicos asociados a diferentes Programas, Departamentos, Divisiones o Secretarías dentro del Ministerio de Salud. Al ingresar el nombre de la oficina, los usuarios pueden obtener una lista de dispositivos, junto con su estado de conexión (activo o inactivo) y el nombre del usuario asignado. Esta funcionalidad es crucial para mantener la eficiencia operativa y la resolución rápida de problemas dentro de la infraestructura de redes.

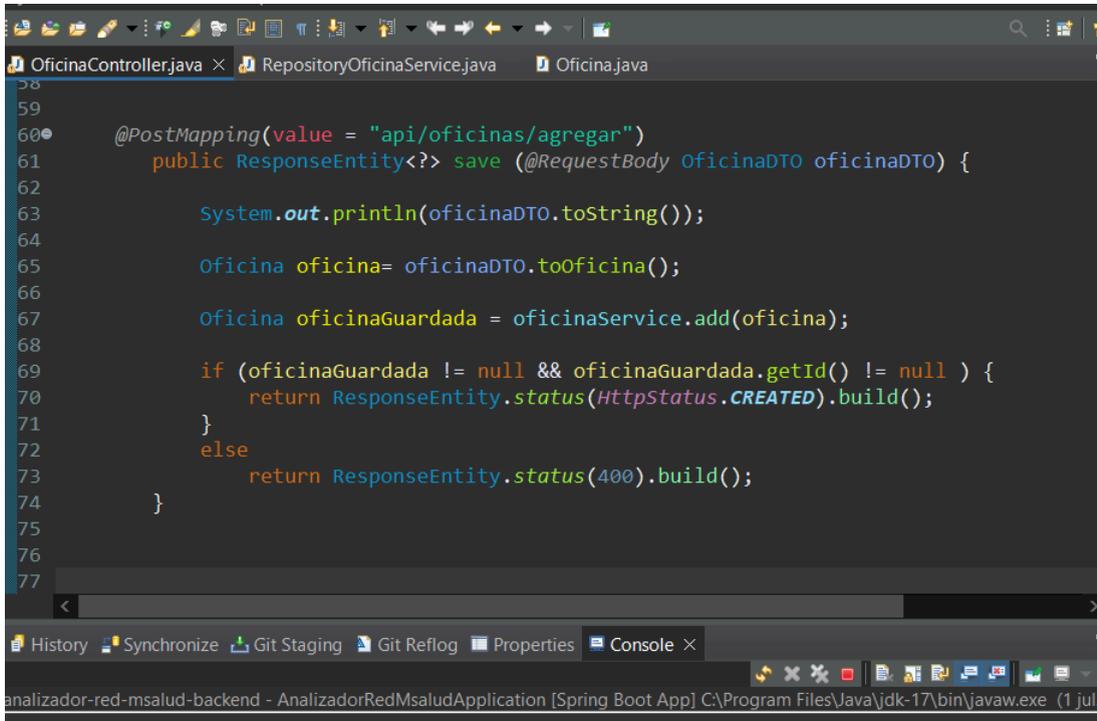
Direccion IP	Estado	Usuario
192.168.0.3	Inactiva	master1002
192.168.100.102	Activa	master1007

Objetivos:

El objetivo principal de la funcionalidad de "Búsqueda por Oficina" es permitir la verificación y el monitoreo centralizado del estado de conectividad de los dispositivos asociados a una oficina específica dentro del Ministerio de Salud. Al proporcionar una visión clara y organizada de los dispositivos por oficina, esta herramienta facilita la identificación rápida de problemas de conectividad, problemas por de cortes de energía, lo que provocan fallos en los en switches, problemas en los puertos switch (boca), detectar problemas mayores en la conexión troncal por ejemplo o una simple placa de red o sistema operativo con fallos

Detalles de Implementación

La funcionalidad de "Búsqueda por Oficina" permite gestionar y monitorizar el estado de los dispositivos dentro de una oficina específica. A continuación, se explica el código de implementación:



```

58
59
60 @PostMapping(value = "api/oficinas/agregar")
61 public ResponseEntity<?> save (@RequestBody OficinaDTO oficinaDTO) {
62
63     System.out.println(oficinaDTO.toString());
64
65     Oficina oficina= oficinaDTO.toOficina();
66
67     Oficina oficinaGuardada = oficinaService.add(oficina);
68
69     if (oficinaGuardada != null && oficinaGuardada.getId() != null ) {
70         return ResponseEntity.status(HttpStatus.CREATED).build();
71     }
72     else
73         return ResponseEntity.status(400).build();
74 }
75
76
77

```

1. Controlador de Oficinas:

- Agregar Oficina:

```

@PostMapping(value = "api/oficinas/agregar")
public ResponseEntity<?> save(@RequestBody OficinaDTO oficinaDTO) {
    Oficina oficina = oficinaDTO.toOficina();
    Oficina oficinaGuardada = oficinaService.add(oficina);

    if (oficinaGuardada != null && oficinaGuardada.getId() != null)
    {
        return ResponseEntity.status(HttpStatus.CREATED).build();
    }
}

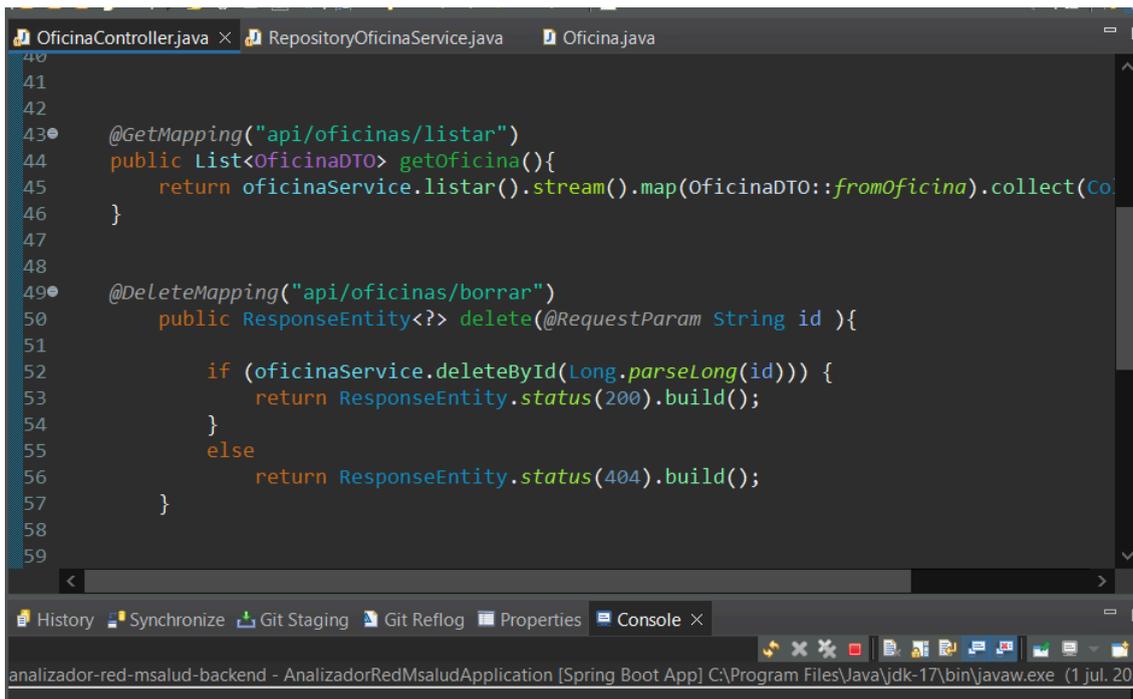
```

```

    } else {
        return ResponseEntity.status(400).build();
    }
}

```

Este método agrega una nueva oficina a la base de datos. Convierte el DTO recibido en una entidad `Oficina`, la guarda mediante el servicio correspondiente y retorna una respuesta HTTP adecuada según el resultado de la operación.



```

OficinaController.java x RepositoryOficinaService.java Oficina.java
40
41
42
43 @GetMapping("api/oficinas/listar")
44 public List<OficinaDTO> getOficina(){
45     return oficinaService.listar().stream().map(OficinaDTO::fromOficina).collect(
46 }
47
48
49 @DeleteMapping("api/oficinas/borrar")
50 public ResponseEntity<?> delete(@RequestParam String id){
51
52     if (oficinaService.deleteById(Long.parseLong(id))) {
53         return ResponseEntity.status(200).build();
54     }
55     else
56         return ResponseEntity.status(404).build();
57 }
58
59
History Synchronize Git Staging Git Reflog Properties Console x
analizador-red-msalud-backend - AnalizadorRedMsaludApplication [Spring Boot App] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 jul. 202

```

- Listar Oficinas:

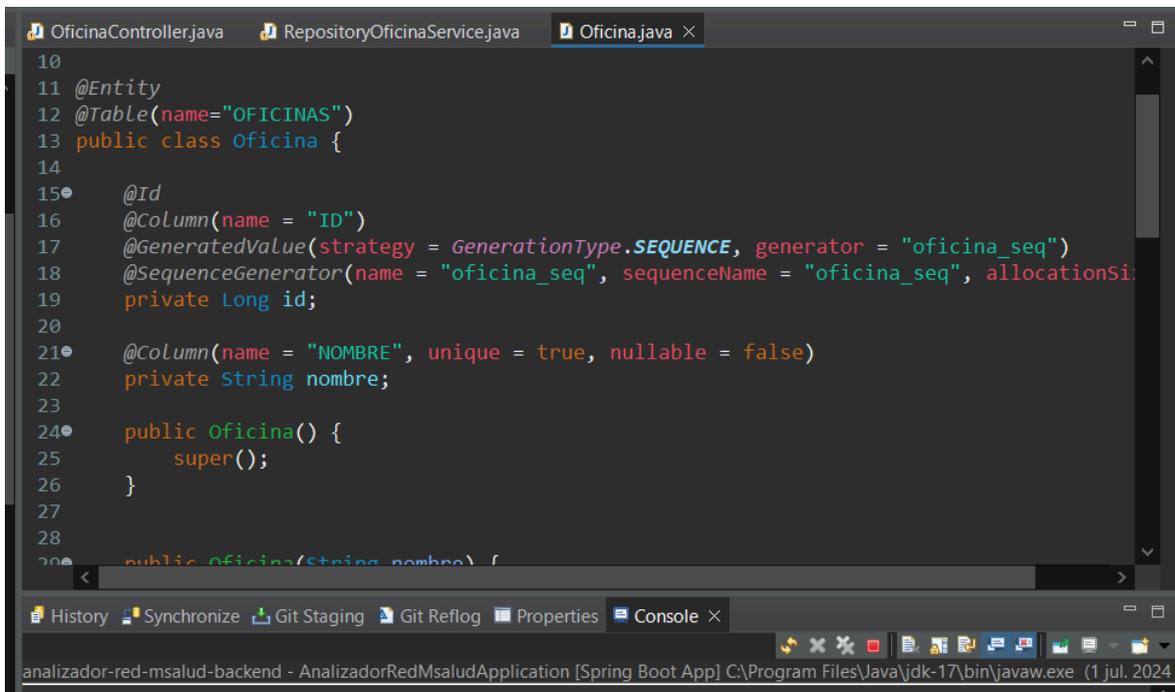
```
@GetMapping("api/oficinas/listar")
public List<OficinaDTO> getOficina() {
    return
oficinaService.listar().stream().map(OficinaDTO::fromOficina).collect(Collectors.toList());
}
```

Este método lista todas las oficinas almacenadas, convirtiendo las entidades `Oficina` a DTOs para su presentación.

- Eliminar Oficina:

```
@DeleteMapping("api/oficinas/borrar")
public ResponseEntity<?> delete(@RequestParam String id) {
    if (oficinaService.deleteById(Long.parseLong(id))) {
        return ResponseEntity.status(200).build();
    } else {
        return ResponseEntity.status(404).build();
    }
}
```

Este método elimina una oficina de la base de datos según su ID. Retorna una respuesta HTTP que indica el éxito o fracaso de la operación.



```

10
11 @Entity
12 @Table(name="OFICINAS")
13 public class Oficina {
14
15     @Id
16     @Column(name = "ID")
17     @GeneratedValue(strategy = GenerationType.SEQUENCE, generator = "oficina_seq")
18     @SequenceGenerator(name = "oficina_seq", sequenceName = "oficina_seq", allocationSi
19     private Long id;
20
21     @Column(name = "NOMBRE", unique = true, nullable = false)
22     private String nombre;
23
24     public Oficina() {
25         super();
26     }
27
28
29     public Oficina(String nombre) {

```

2. Modelo de Oficina:

```

@Entity
@Table(name = "OFICINAS")
public class Oficina {
    @Id
    @GeneratedValue(strategy = GenerationType.SEQUENCE, generator =
"oficina_seq")
    @SequenceGenerator(name = "oficina_seq", sequenceName =
"oficina_seq", allocationSize = 1)
    @Column(name = "ID")
    private Long id;

    @Column(name = "NOMBRE", unique = true, nullable = false)
    private String nombre;

    public Oficina() { }

```

```
public Oficina(String nombre) {  
    this.nombre = nombre;  
}  
  
// Getters y setters...  
  
}
```

Esta clase define la entidad `Oficina`, con sus propiedades y anotaciones JPA para mapearla a la base de datos. Se puede observar el parametro “unique=true” impide que se duplique la creación de oficinas con el mismo nombre.

La funcionalidad de "Búsqueda por Oficina" permite agregar, listar y eliminar oficinas. Utiliza Spring Boot para manejar las solicitudes HTTP y JPA para interactuar con la base de datos, proporcionando una solución para el monitoreo de dispositivos por oficina.

Funcionalidad ABM Usuarios

La funcionalidad de ABM (Alta, Baja, Modificación) Usuarios fue desarrollada para gestionar los usuarios



del sistema, facilitando la administración de sus datos y la supervisión de su actividad dentro de la red del Ministerio de Salud.

Objetivo

El objetivo principal de la funcionalidad ABM Usuarios es proporcionar una herramienta centralizada para agregar, eliminar y modificar usuarios, mejorando la gestión y el control de acceso a los recursos tecnológicos del Ministerio. Esta funcionalidad asegura que los datos de los usuarios estén actualizados y que se pueda supervisar su actividad en la red.

Modo de Uso

1. Alta de Usuario:

- El usuario administrador navega a la sección "ABM Usuarios" y selecciona la opción "Agregar".
- Completa el formulario con los datos del nuevo usuario, incluyendo DNI, Nombre Master, Oficina y Dirección IP.
- Hace clic en "Crear" para registrar el nuevo usuario en el sistema.

2. Modificación de Usuario:

- En la lista de usuarios, el administrador selecciona el usuario a modificar y hace clic en el icono de edición.
- Actualiza los campos necesarios en el formulario de edición.
- Hace clic en "Guardar" para aplicar los cambios.

3. Baja de Usuario:

- En la lista de usuarios, el administrador selecciona el usuario a eliminar y hace clic en el icono de eliminación (tacho de basura).
- Confirma la acción para eliminar el usuario del sistema.

Detalles de Implementación

```

OficinaController.java  RepositoryOficinaService.java  Oficina.java  UsuarioController.java x
25  @Qualifier("repositoryOficinaService")
26  private OficinaService oficinaService;
27
28
29  @GetMapping("/api/usuarios/listar")
30  public List<UsuarioDTO> getUsuarios(){
31      System.out.println(usuarioService.listar().stream().map(UsuarioDTO::fromUsuario).collect(Collectors.toList()));
32      return usuarioService.listar().stream().map(UsuarioDTO::fromUsuario).collect(Collectors.toList());
33  }
34
35  @GetMapping("/api/usuarios/busquedaXOficina")
36  public List<UsuarioDTO> getBusquedaPorOficina(@RequestParam Long id_oficina) {
37      return this.usuarioService.buscarXOficina(id_oficina);
38      //System.out.println(this.usuarioService.buscarXOficina(oficina));
39  }
40
41  @GetMapping("/api/usuarios/getById")
42  public ResponseEntity<> getUsuarioById(@RequestParam Long id) {
43      try {
44          Usuario unUsuario = this.usuarioService.get(id);
45          if (unUsuario == null) {
46              return new ResponseEntity<>("Usuario no encontrado", HttpStatus.NOT_FOUND);
47          }
48          UsuarioDTO usuarioDTO = UsuarioDTO.fromUsuario(unUsuario);
49          return new ResponseEntity<>(usuarioDTO, HttpStatus.OK);
50      } catch (Exception e) {
51          return new ResponseEntity<>("Error al obtener el usuario: " + e.getMessage(), HttpStatus.INTERNAL_SERVER_ERROR);
52      }
53  }
54
55

```

```

54
55
56  @PostMapping(value = "api/usuarios/agregar")
57  public ResponseEntity<> save (@RequestBody UsuarioDTO usuarioDTO) {
58
59      System.out.println(usuarioDTO);
60
61      Usuario usuario = usuarioDTO.toUsuario();
62
63      if (!usuarioService.existsByName(usuario.getNombre_master())) {
64          usuarioService.save(usuario);
65          return ResponseEntity.status(HttpStatus.CREATED).build();
66      }
67      else
68          return ResponseEntity.status(HttpStatus.BAD_REQUEST).build();
69  }
70
71  @PutMapping("api/usuarios/editar")
72  public ResponseEntity<> editar (@RequestBody UsuarioDTO usuarioDTO) {
73      Usuario usuario = usuarioDTO.toUsuario();
74
75      if ((usuarioService.get(usuario.getId())) != null) {
76          usuarioService.save(usuario);
77          return ResponseEntity.status(204).body("Usuario modificado correctamente");
78      }
79      else
80          return ResponseEntity.status(400).body("El usuario NO existe");
81  }
82

```

```
82
83 @DeleteMapping("api/usuarios/borrar")
84 public ResponseEntity<> delete(@RequestParam String id ){
85
86     if (usuarioService.deleteById(Long.parseLong(id))) {
87         return ResponseEntity.status(200).build();
88     }
89     else
90         return ResponseEntity.status(404).build();
91 }
92
93
94
95
96 }
97
```

History Synchronize Git Staging Git Reflog Properties Console ×

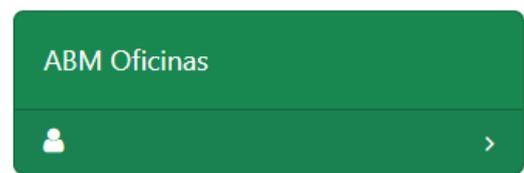
analizador-red-msalud-backend - AnalizadorRedMsaludApplication [Spring Boot App] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 jul. 2024
[Usuario [id=1, dni=30555001, nombre_master=master1000, oficina=Oficina [id=1, nombre=salud mental], ip=/192.168.0.1, ultima_conexior

1. Listar Usuarios: Recupera todos los usuarios almacenados en la base de datos y los convierte a DTOs para su presentación en la interfaz de usuario.
2. Agregar Usuario: Recibe los datos de un nuevo usuario a través de un DTO, los convierte a una entidad `Usuario` y los guarda en la base de datos.
3. Editar Usuario: Permite actualizar los datos de un usuario existente, asegurando que los cambios se reflejan en la base de datos.
4. Eliminar Usuario: Elimina un usuario específico de la base de datos, identificado por su ID.

Esta funcionalidad permite una gestión integral de los usuarios, asegurando que los datos estén siempre actualizados y proporcionando herramientas para mantener un control efectivo sobre la actividad en la red del Ministerio de Salud.

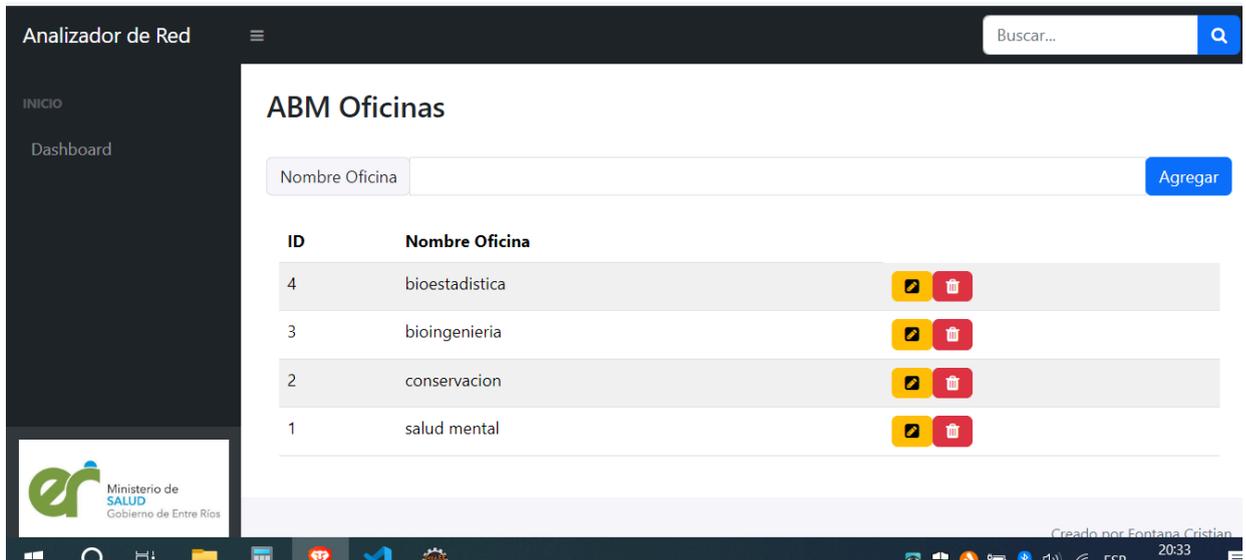
Funcionalidad de ABM Oficinas

La funcionalidad de ABM (Alta, Baja, Modificación) Oficinas fue desarrollada para gestionar de manera eficiente las oficinas dentro del sistema, facilitando la administración de sus datos y la organización de recursos tecnológicos en diferentes áreas del Ministerio de Salud.



Objetivo

El objetivo principal de la funcionalidad ABM Oficinas es proporcionar una herramienta centralizada para agregar, eliminar y modificar oficinas, mejorando la gestión y el control de los recursos tecnológicos distribuidos en diversas ubicaciones.



Modo de Uso

1. Alta de Oficina:

- El usuario administrador navega a la sección "ABM Oficinas" y selecciona la opción "Agregar".
- Completa el campo con el nombre de la nueva oficina.
- Hacer clic en "Agregar" para registrar la nueva oficina en el sistema.

2. Modificación de Oficina:

- En la lista de oficinas, el administrador selecciona la oficina a modificar y hace clic en el icono de edición.
- Actualiza el nombre de la oficina en el formulario de edición.
- Hace clic en "Guardar" para aplicar los cambios.

3. Baja de Oficina:

- En la lista de oficinas, el administrador selecciona la oficina a eliminar y hace clic en el icono de eliminación (tacho de residuos).
- Confirma la acción para eliminar la oficina del sistema.

Detalles de Implementación

```

41
42
43 @GetMapping("api/oficinas/listar")
44 public List<OficinaDTO> getOficina(){
45     return oficinaService.listar().stream().map(OficinaDTO::fromOficina).collect(Collectors.toList());
46 }
47
48
49 @DeleteMapping("api/oficinas/borrar")
50 public ResponseEntity<?> delete(@RequestParam String id ){
51
52     if (oficinaService.deleteById(Long.parseLong(id))) {
53         return ResponseEntity.status(200).build();
54     }
55     else
56         return ResponseEntity.status(404).build();
57 }
58
59
60 @PostMapping(value = "api/oficinas/agregar")
61 public ResponseEntity<?> save (@RequestBody OficinaDTO oficinaDTO) {
62
63     System.out.println(oficinaDTO.toString());
64
65     Oficina oficina= oficinaDTO.toOficina();
66
67     Oficina oficinaGuardada = oficinaService.add(oficina);
68
69     if (oficinaGuardada != null && oficinaGuardada.getId() != null ) {
70         return ResponseEntity.status(HttpStatus.CREATED).build();
71     }

```

analizador-red-msalud-backend - AnalizadorRedMsaludApplication [Spring Boot App] C:\Program Files\Java\jdk-17\bin\javaw.exe (1 jul. 2024) [Usuario [id=1, dni=30555001, nombre_master=master1000, oficina=Oficina [id=1, nombre=salud mental], ip=/192.168.0.1, ultima_conexion=...]

Explicación del Código y Funcionalidades

1. Listar Oficinas: Recupera todas las oficinas almacenadas en la base de datos y las convierte a DTOs para su presentación en la interfaz de usuario.
2. Agregar Oficina: Recibe los datos de una nueva oficina a través de un DTO, los convierte a una entidad `Oficina` y los guarda en la base de datos.

3. Eliminar Oficina: Elimina una oficina específica de la base de datos, identificada por su ID.

Esta funcionalidad permite una gestión integral de las oficinas, asegurando que los datos estén siempre actualizados y proporcionando herramientas para mantener un control sobre la distribución y organización de los dispositivos del Ministerio de Salud.

Conclusiones

Aprendizajes obtenidos en la realización de la práctica

Durante la práctica profesional, uno de los aprendizajes más valiosos fue comprender la importancia y el poder de utilizar JPA (Java Persistence API) con Hibernate. Cuando se me propuso ,a mitad de camino ,cambiar el motor de base de datos de PostgreSQL a MySQL, inicialmente pensé que sería un proceso largo y complicado. Sin embargo, gracias a JPA y Hibernate, descubrí que este cambio se podía realizar con solo unas simples líneas de configuración. Este conocimiento no solo me permitió ahorrar tiempo, sino que también

mejoró mi comprensión sobre la flexibilidad y adaptabilidad de estas herramientas en proyectos reales.

Además, aprendí la importancia de entender y comunicarse con las personas que utilizarán el sistema. Al conversar con los usuarios finales, pude interpretar sus necesidades y expectativas, lo que me ayudó a diseñar un sistema más alineado con sus intereses. Este proceso de interacción me enseñó a valorar la retroalimentación y a utilizarla para mejorar continuamente el sistema.

En resumen, esta experiencia me proporcionó habilidades técnicas y sociales fundamentales para el desarrollo de software, enseñándome a mejorar en lo personal al perder el miedo y la vergüenza que uno tiene muchas veces al salir fuera de su ámbito laboral.

Comentarios personales del trabajo realizado

Estoy realmente contento con el resultado del desarrollo de la aplicación web. Creo que quedó muy bien, con una interfaz agradable a la vista y fácil de usar. Además, fue bien

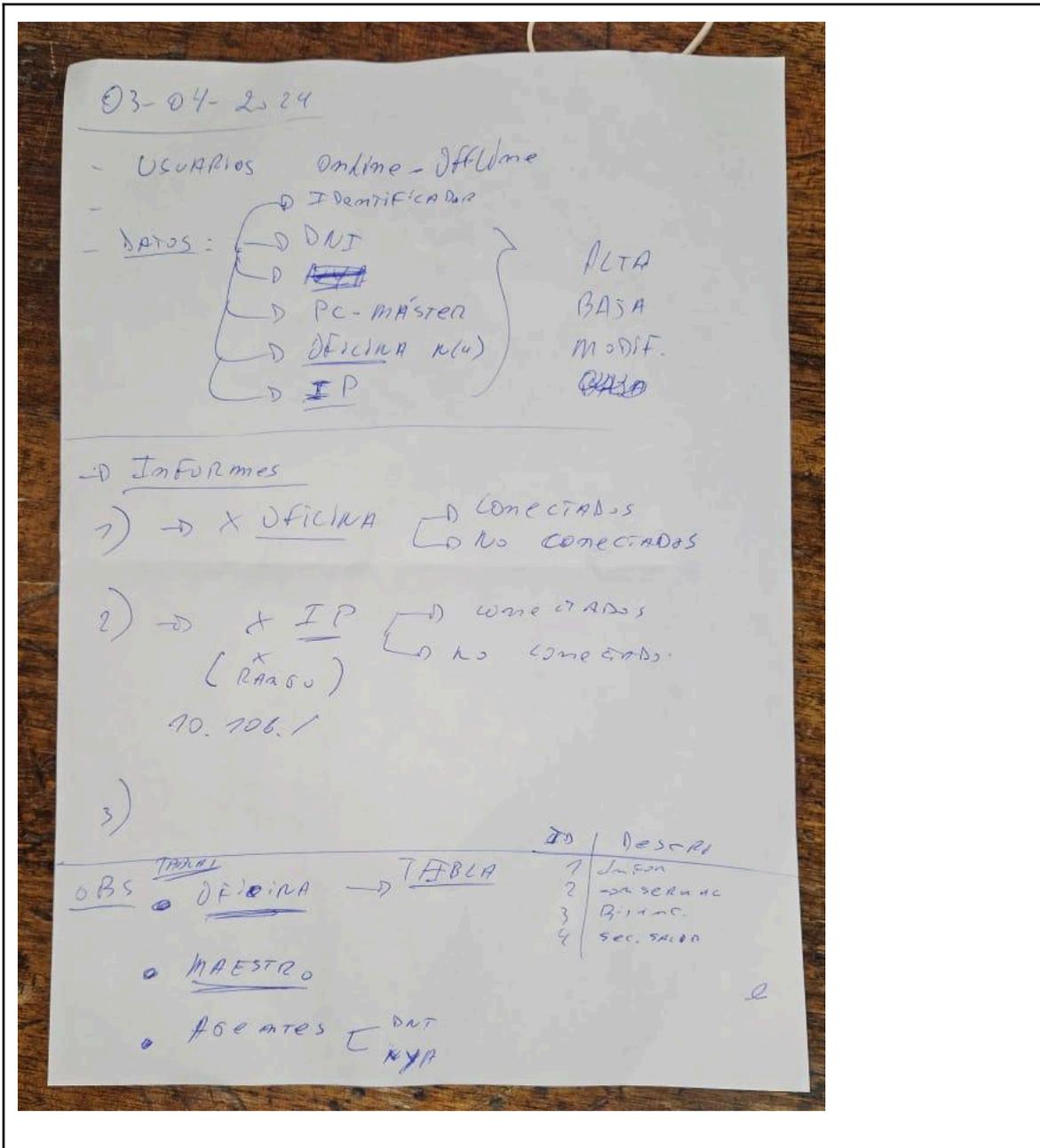
recibida por el personal del departamento, lo que me dejó muy satisfecho. Por supuesto, siempre hay espacio para mejoras y tengo muchas ideas pendientes por implementar. La posibilidad de seguir mejorando la aplicación día a día es algo que me motiva y me hace ver que este es solo el comienzo de un proyecto con mucho potencial.

Conclusiones Generales

La realización de esta práctica profesional ha sido una experiencia enriquecedora tanto a nivel técnico como personal. La implementación de la aplicación web no solo mejoró la gestión y monitoreo de la red informática del Ministerio de Salud. Además, el proceso de interacción con los usuarios finales resaltó la importancia de comprender y atender sus necesidades e inquietudes sobre el nuevo sistema, lo que resultó en un producto bien recibido y útil para el departamento informática. Este proyecto es una muestra de que siempre hay oportunidades de mejora y crecimiento.

Estoy entusiasmado por las futuras actualizaciones y mejoras que se puedan implementar, en la práctica no solo cumplió con los objetivos establecidos, sino que también abrió nuevas perspectivas y desafíos para el desarrollo continuo.

Anexos

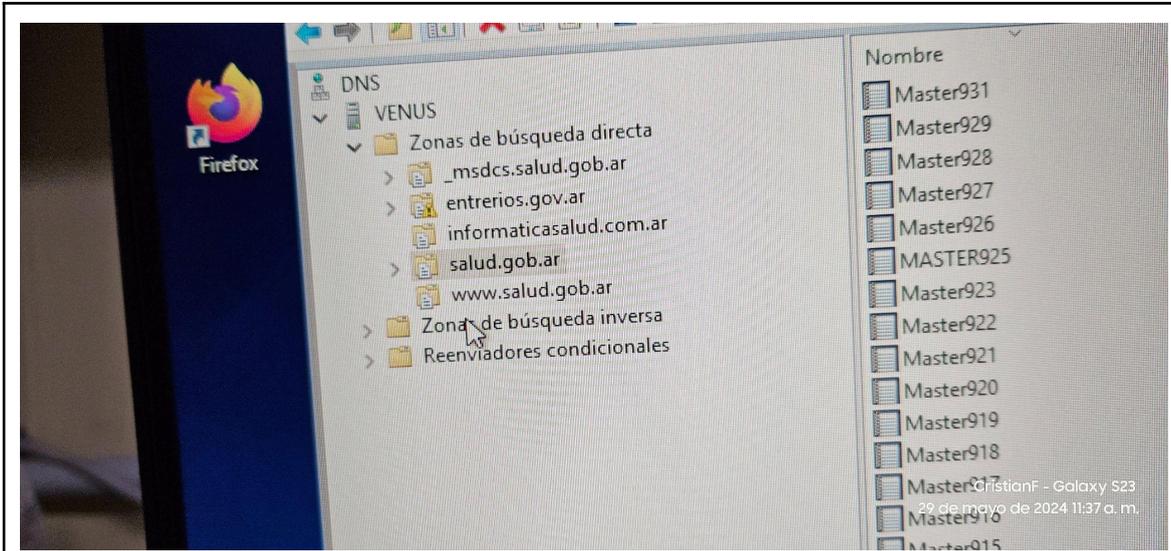


Instructivo creado por el tutor de la pasantía (Dario Reisenauer) con el diagramado de las actividades a realizar en la pasantía

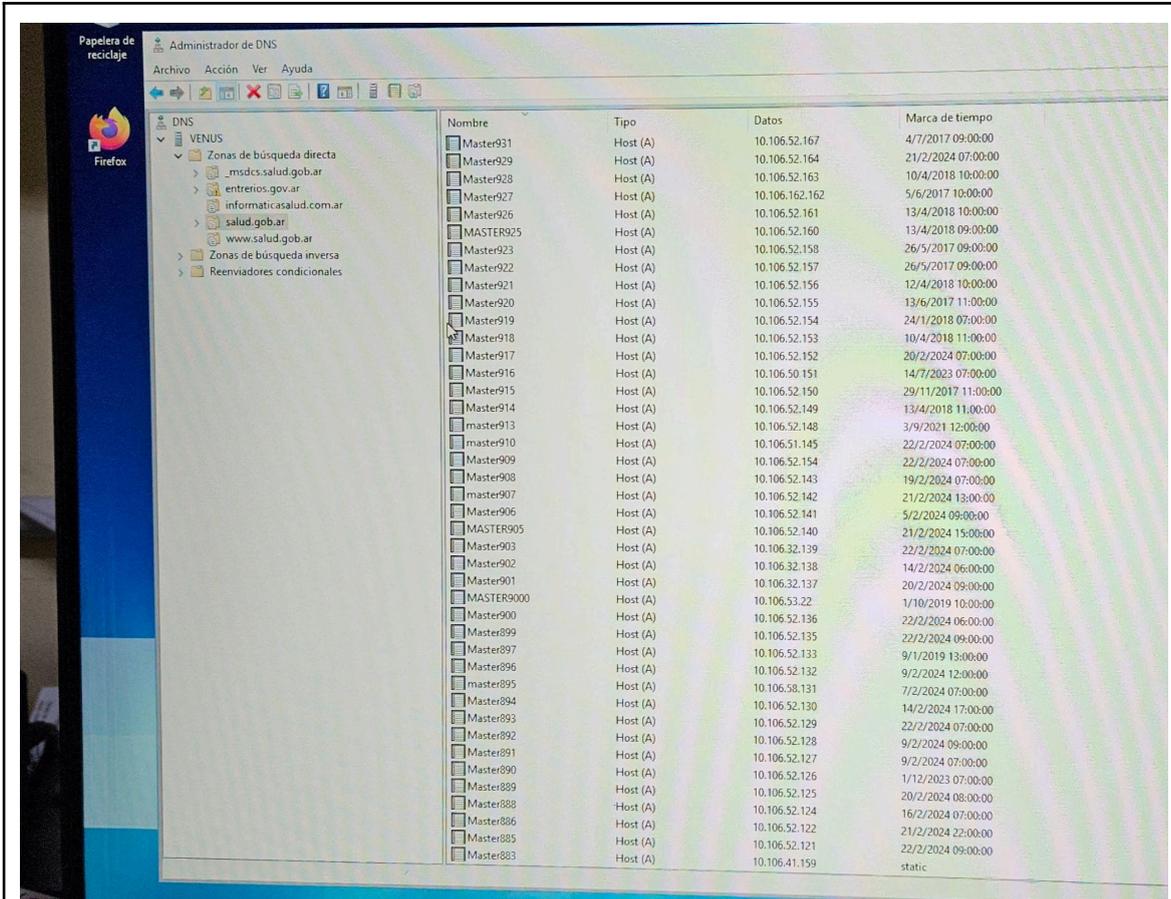
Listado de Teléfonos - Secretaría de Salud

Nro.	Oficina	Nro.	Oficina
420-9600	OPERADORA 1	420-9661	FARMACIA (Biblioteca-Audi.)
420-9650	OPERADORA 2	420-9684	GUARDERÍA
420-9621	ADMINISTRACIÓN (Secretaria)	420-9662	GUARDIA POLICIAL
420-9605	ADMINISTRACIÓN - DIRECCION - FAX	420-9634	INFORMATICA
420-9660	ANTICIPOS DE FONDOS	420-9627	LIQUIDACIONES (M.de entradas - Embargos) <i>420-9662</i>
420-9647	ARANCELAMIENTO	420-9648	LIQUIDACIONES (Suplencias - Jefatura)
420-9601	ASESORIA LEGAL	420-9659	MANTENIMIENTO
420-9607	ATENCION MEDICA (Concursos)	420-9635	MATERNIDAD E INFANCIA
420-9628	ATENCION MEDICA (Contralor Profesional)	420-9637	MESA DE ENTRADAS (Jefatura)
422-0137	ATENCION MEDICA (Departamento)	420-8814	MINISTRO Despacho
420-9625	ATENCION MEDICA DIRECCION (Gestión Adm.)	420-8806	MINISTRO Privada
420-9630	ATENCION MEDICA DIRECCION (Gestión Adm.)	420-9613	ODONTOLOGIA
420-9653	ATENCION PRIMARIA	424-8566	ODONTOLOGIA (Taller)
420-9602	ATENCION PRIMARIA	431-3664	PLAN NACER
420-7968	ATENCION PRIMARIA (GESTION ADMINISTRATIVA)	420-9654	PLANIFICACION
420-9682	AUTOMOTORES (Calle Perú)	420-9655	PRENSA Y DIFUSION
422-4882	BANCO DE SANGRE (Raquel)	420-9644	PRESUPUESTO
420-7804	BIOESTADISTICAS	420-7968	PROG. ESPECIALES
420-9612	BIOINGENIERIA	423-0649	PROG. ESPECIALES (Dir.)
420-9608	CAPACITACION Y DOCENCIA	420-9626	PROG.CANCER DE CUELLO-BIOQUIMICA-BIBLIOTECA
431-0028	CLINICA ESCOLAR (Sta. Fe y 25 de Junio)	420-7969	PROGRAMA FEDERAL DE SALUD
420-9619	CONSERVACION	420-9658	PROGRAMA VIOLENCIA (DIR. SAL. MENTAL en Clínica Esc.)
420-9643	CONTABILIDAD (Gustavo Irigoitia)	420-9617	RECURSOS HUMANOS (Jefatura)
420-9646	CONTRATAIONES (Patricia Martínez)	420-9636	RECURSOS HUMANOS (Lic. - Partes diarios)
420-9641	CONTROL DEL GASTO	422-7565	RECURSOS HUMANOS (Salarios)
484-0355	COORDINACION GRAL DE MINISTERIO (Dr. Rodriguez)	423-5766	RECURSOS HUMANOS (Suplencias) <i>HAGA</i>
420-9656	DESPACHO	420-9645	RENDICIONES DE CUENTAS
420-9614	DESPACHO	420-9631	RESIDENCIAS MEDICAS
420-9603	DESPACHO (Tramitación)	420-9610	SALUD MENTAL (Dirección)
420-9609	DESPACHO (Dirección)	420-9620	SECRETARIA DE GESTION SANITARIA (Privada Dr. Basso)
484-0731	DIAGNOSTICO POR IMG.	420-9604	SECRETARIA PRIVADA FAX
484-0210	DIREC. MUNICIPIOS Y COMUNIDADES SALUDABLES (Dr. Roko)	420-9692	SECRETARIA PRIVADA (Asesor)
420-9681	DIRECCION DE DEFENSA CIVIL	420-9670	SUBSECRETARIA DE ADMINISTRACION
420-7818/19	DIRECCION DE RECURSOS HUMANOS	420-9686	SUBSECRETARIA DE ADMINISTRACION (Subsec. <i>SECRETARIA</i>)
420-9680	EMERGENCIA (Calle Perú)	420-9611	SUBSECRETARIA DE SER. AS. Y G. (Privada)
420-9683	EMERGENCIA (Comunicaciones Calle Gychú.)	420-9622	SUBSECRETARIA PROMOCION Y PREVENCIÓN (Dra Leiva)
420-9685	EMERGENCIA SANITARIA (Director)	420-9638	TESORERIA (Jefatura)
420-9629	ENFERMERIA	420-8870/71	UCC
420-9652	EPIDEMIOLOGIA <i>420-9629 (Dra. E. Fontana)</i>	420-8874	UCC Arquelles
420-8803	EPIDEMIOLOGIA (Laboratorio)	420-9663	VIATICOS - ANTICIPO DE FONDOS
420-9651	EXT COMUNIDAD - PROMAPS	420-9690	<i>420-9661 420-9690</i>

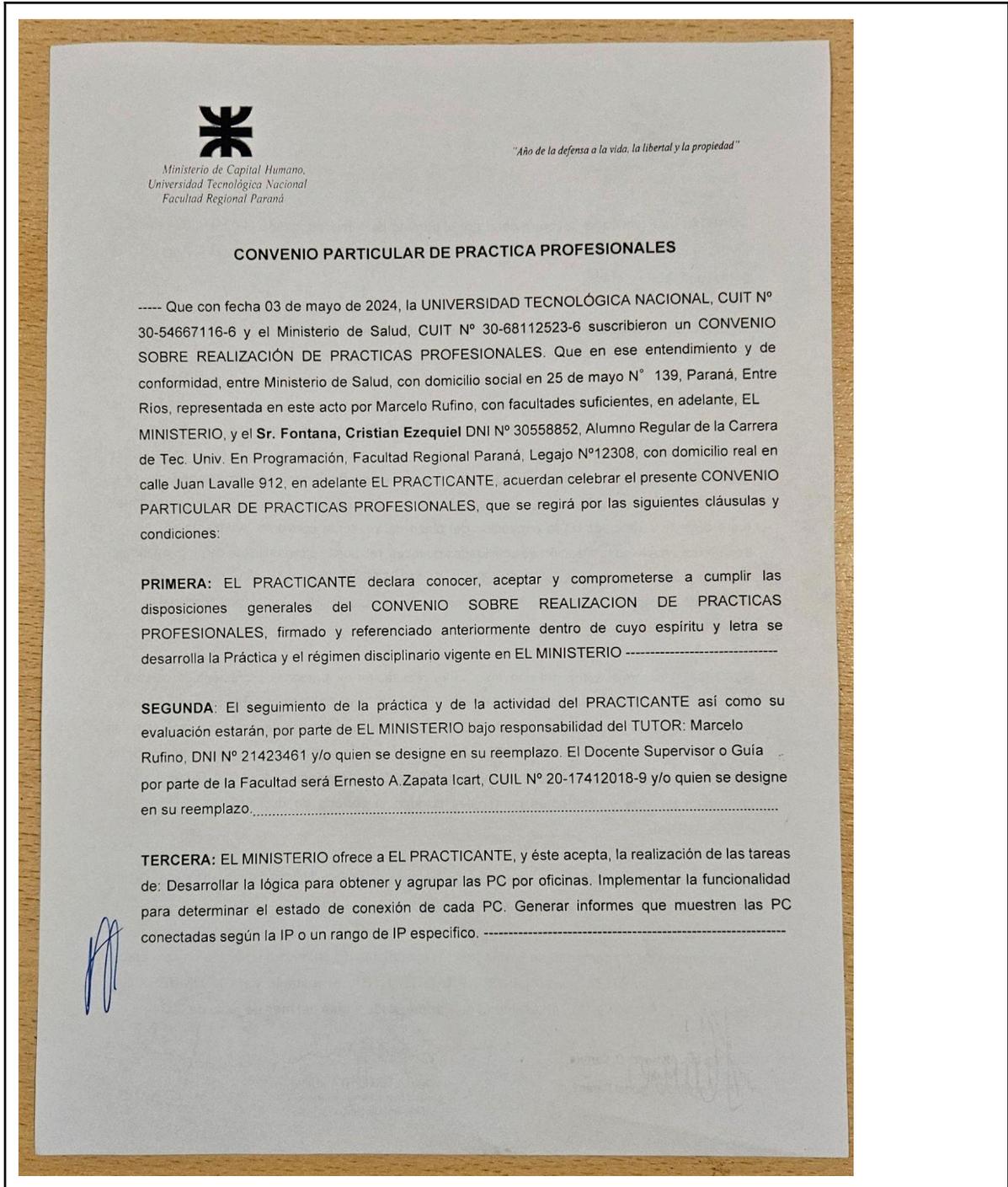
Listado de oficinas del Ministerio de Salud

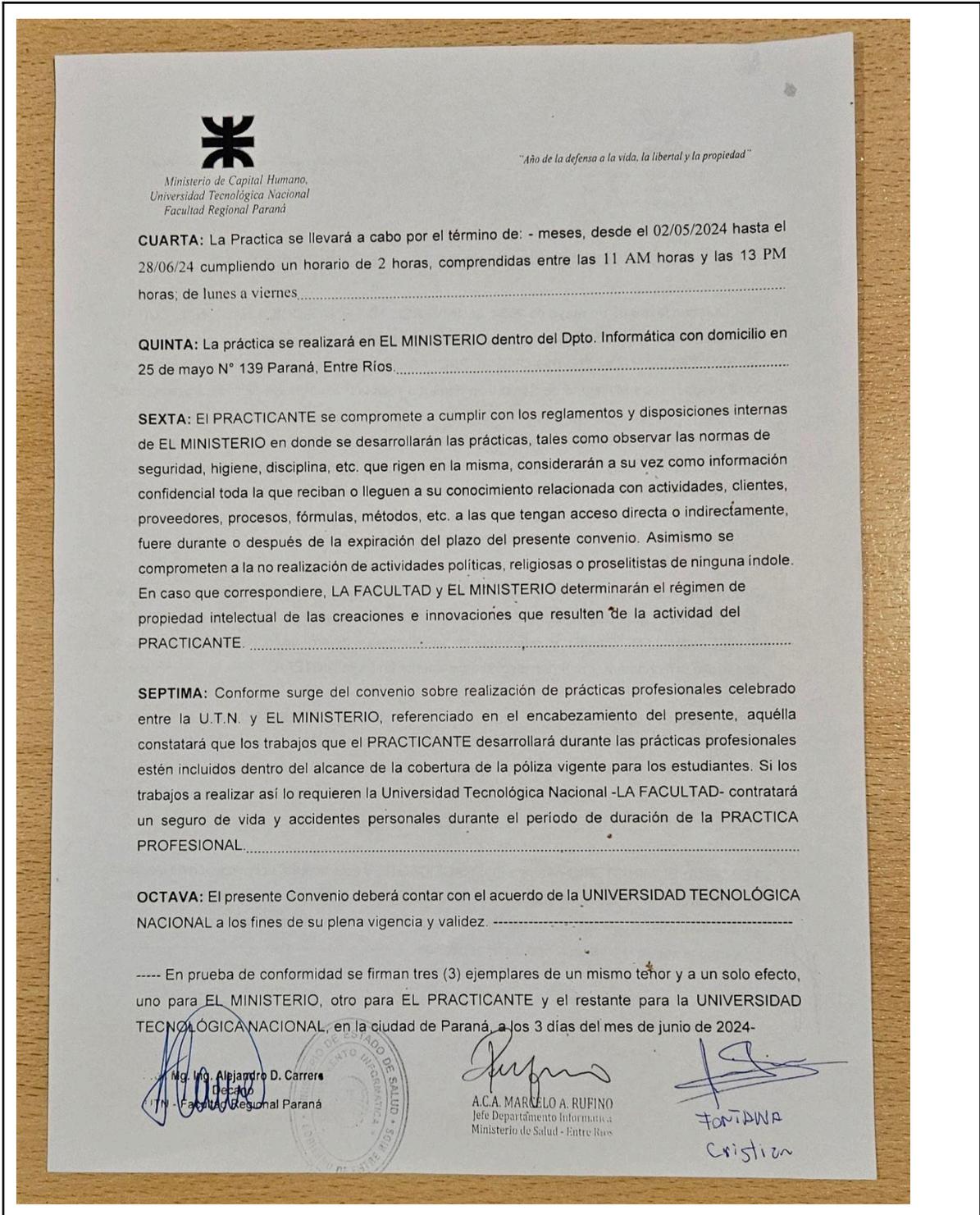


Listado de distribución de Active Directory



Muestra parcial de usuarios registrados del Servidor Active Directory.






Ministerio de Capital Humano,
Universidad Tecnológica Nacional
Facultad Regional Paraná

"Año de la defensa a la vida, la libertad y la propiedad"

CUARTA: La Practica se llevará a cabo por el término de: - meses, desde el 02/05/2024 hasta el 28/06/24 cumpliendo un horario de 2 horas, comprendidas entre las 11 AM horas y las 13 PM horas, de lunes a viernes.....

QUINTA: La práctica se realizará en EL MINISTERIO dentro del Dpto. Informática con domicilio en 25 de mayo N° 139 Paraná, Entre Ríos.....

SEXTA: El PRACTICANTE se compromete a cumplir con los reglamentos y disposiciones internas de EL MINISTERIO en donde se desarrollarán las prácticas, tales como observar las normas de seguridad, higiene, disciplina, etc. que rigen en la misma, considerarán a su vez como información confidencial toda la que reciban o lleguen a su conocimiento relacionada con actividades, clientes, proveedores, procesos, fórmulas, métodos, etc. a las que tengan acceso directa o indirectamente, fuere durante o después de la expiración del plazo del presente convenio. Asimismo se comprometen a la no realización de actividades políticas, religiosas o proselitistas de ninguna índole. En caso que correspondiere, LA FACULTAD y EL MINISTERIO determinarán el régimen de propiedad intelectual de las creaciones e innovaciones que resulten de la actividad del PRACTICANTE.....

SEPTIMA: Conforme surge del convenio sobre realización de prácticas profesionales celebrado entre la U.T.N. y EL MINISTERIO, referenciado en el encabezamiento del presente, aquélla constatará que los trabajos que el PRACTICANTE desarrollará durante las prácticas profesionales estén incluidos dentro del alcance de la cobertura de la póliza vigente para los estudiantes. Si los trabajos a realizar así lo requieren la Universidad Tecnológica Nacional -LA FACULTAD- contratará un seguro de vida y accidentes personales durante el período de duración de la PRACTICA PROFESIONAL.....

OCTAVA: El presente Convenio deberá contar con el acuerdo de la UNIVERSIDAD TECNOLÓGICA NACIONAL a los fines de su plena vigencia y validez.

----- En prueba de conformidad se firman tres (3) ejemplares de un mismo tenor y a un solo efecto, uno para EL MINISTERIO, otro para EL PRACTICANTE y el restante para la UNIVERSIDAD TECNOLÓGICA NACIONAL, en la ciudad de Paraná, a los 3 días del mes de junio de 2024-

Mg. Ing. Alejandro D. Carrere
Decano
UTN - Facultad Regional Paraná

A.C.A. MARCELO A. RUFINO
Jefe Departamento Informática
Ministerio de Salud - Entre Ríos

Fontana
Cristian

Convenio UTN - Ministerio de salud - Pasantía Fontana Cristian Ezequiel