Aplicación: Monitoreo de impresoras

El presente trabajo consiste en una aplicación escrita en el lenguaje Python 3 que tiene como finalidad el monitoreo de ciertas características de impresoras de red en entornos empresariales.

**Utilidad**

Vemos una gran utilidad de esta aplicación siendo ejecutada en estos entornos empresariales que cuentan con múltiples impresoras de red ya que representa una automatización que no requiere de intervención humana alguna en su normal funcionamiento ya que la misma se ejecuta en segundo plano realizando consultas a cada una de las impresoras involucradas cada cierto tiempo (Ese tiempo fue configurado como 1 minuto, aunque puede ser modificado a gusto). Al detectar alguna variable dentro de cierto umbral se envía un correo electrónico de aviso

**Requerimientos**

La puesta a punto de esta consiste en agregar manualmente cada impresora con sus respectivos datos de conexión, como dirección IP, puerto, protocolo a usar (http o https), marca, y la dirección interna en la cual se encuentra el archivo de mantenimiento de esta. Estos datos deben ser especificados en el archivo “devices.json” que se encuentra en el directorio “app”

Además, se necesita agregar un correo electrónico para enviar las alertas y otro para recibirlas (Nota: Puede usarse el mismo para enviar y recibirlas).

Se diseño un sistema alternativo para cuando se presentan problemas en cuanto a la cuota máxima de correos enviados por cuenta y por día. Esto hace que pueda configurarse además otra cuenta de correo electrónico, llamada “primer backup”. Estos datos deben ser especificados en el archivo “config.json” que se encuentra en el directorio “app”

**Plataforma testeada**

Se han llevado a cabo pruebas en impresoras de la marca Brother, modelos HL-1210W y DCP-1610NW. Se ha diseñado una estructura tal para poder agregar otras marcas de manera intuitiva dentro del directorio “supported\_brands”. Actualmente, dentro de este se encuentra un archivo llamado “brother.json” que hace referencia a las variables a medir con respecto a esta marca.

**Características a monitorear**

En cuanto a las características a ser monitoreadas se encuentran: el tóner disponible en la impresora (en porcentaje) y la vida útil del tambor (en términos absolutos y porcentaje)

En las impresoras Brother, "Remaining Life (Drum Unit)" y "% of Life Remaining (Drum Unit)" se refieren a dos formas diferentes de medir la vida útil restante del tambor de la impresora.

"Remaining Life (Drum Unit)" se refiere a la cantidad de vida útil del tambor que queda en términos absolutos. Por lo general, se muestra como una cifra en páginas impresas o en un porcentaje que indica el número estimado de páginas que se pueden imprimir antes de que sea necesario reemplazar el tambor. Por ejemplo, si muestra "Remaining Life (Drum Unit): 2000 páginas", significa que el tambor puede imprimir aproximadamente 2000 páginas adicionales antes de que sea recomendable cambiarlo.

"% of Life Remaining (Drum Unit)" es una medida relativa que muestra el porcentaje de vida útil que aún queda en el tambor. Por ejemplo, si muestra "% of Life Remaining (Drum Unit): 50%", significa que el tambor ha utilizado aproximadamente la mitad de su vida útil y le queda el 50% restante antes de que sea necesario reemplazarlo.

Ambas medidas son útiles para estimar la vida útil del tambor de la impresora. Algunas impresoras Brother pueden mostrar solo una de estas medidas, mientras que otras pueden mostrar ambas para proporcionar una mejor comprensión del estado del tambor. En general, es recomendable reemplazar el tambor cuando su vida útil restante es baja, ya sea en términos absolutos (Remaining Life) o en porcentaje (% of Life Remaining), para asegurar una calidad de impresión óptima.

**Sistema de alertas**

La aplicación usa un sistema de clasificación de 3 niveles: WARNING (advertencia), CRITICAL (crítico) y ERROR (error) para tener en cuenta a la hora de clasificar los estados de las variables medidas.

* Para el nivel de tóner disponible:
* Se usa un porcentaje que va de 0 % a 100 %
* Un nivel de tóner entre 11 y 50 es categorizado como WARNING
* Un nivel de tóner entre 1 y 10 es categorizado como CRITICAL
* Un nivel de tóner exactamente en 0 es categorizado como ERROR
* Niveles superiores a 50 no son categorizados
* Para la vida útil del tambor (en porcentaje):
* El sistema funciona igual al anteriormente descripto
* Para la vida útil del tambor (en términos absolutos):
* Se usa un valor real entre 0 y 15 000 (15 000 es el dato oficial de vida útil proporcionado por Brother para la vida útil del tambor)
* Un nivel de entre 1501 y 7500 es categorizado como WARNING
* Un nivel de entre 1 y 1 500 es categorizado como CRITICAL
* Un nivel de exactamente 0 es categorizado como ERROR
* Niveles superiores a 7 500 no son categorizados

**Ejecución**

La aplicación se ejecuta usando contenedores Docker. Esto lleva a un fácil despliegue en cualquier sistema operativo disponible evitando tener problemas de compatibilidad con respecto a librerías y sistemas operativos. Además, gracias a que fue escrito usando Docker Compose se puede realizar el despliegue con tan sólo un comando: “*docker-compose up -d*” ubicándonos en la terminal en la raíz del proyecto

Cabe mencionar que se diseñó un sistema de conexión en paralelo con respecto a todos los equipos de red para acortar tiempos de espera.

Además, se diseñó otro sistema de envío de correos electrónicos en paralelo por el mismo motivo.

Ambas funcionalidades hacen que los tiempos de demora de la aplicación se acorten bastante comparándolos a la situación en que la aplicación funcionara de manera secuencial. Recordemos que esta aplicación fue pensada para incluir múltiples impresoras de red y dado el caso enviar múltiples correos electrónicos de aviso, razones por las cuales se justifica implementar ejecución de código en paralelo usando hilos.

**Funcionamiento**

La aplicación se ejecuta por default cada 1 minuto conectándose a los equipos incluidos en el archivo *devices.json*. Una vez realizada la conexión, se descarga por medio de la librería requests un archivo de mantenimiento en formato CSV que tienen las impresoras.   
 Luego de eso, se hace un análisis de ese archivo y se obtienen los valores de los atributos a ser monitoreados. Posterior a eso, se cargan los valores límites (especificados en *config.json*) y se comparan contra los valores actuales que hay dentro de ese archivo CSV.

Sólo se envía un correo electrónico en el caso en que corresponda. La aplicación se cierra y vuelve a ejecutarse al próximo minuto.

El valor de 1 minuto puede ser modificado en el archivo *docker-entrypoint.sh.* Alli, aparece un valor de 60 (correspondiente a 60 segundos)