

# Informe sobre el Proceso de Envío de Correos de Felicitación de Cumpleaños

## 1. Introducción

Este informe describe el proceso automatizado desarrollado para el envío de correos de felicitación de cumpleaños a los empleados de una organización. El proceso se basa en la ejecución de un script Batch que itera sobre una lista de cumpleaños almacenada en un archivo CSV y, si la fecha de hoy coincide con el cumpleaños de un empleado, se envía un correo personalizado utilizando un script de Python.

## 2. Descripción del Proceso

### 2.1 Lectura del Archivo CSV

El archivo CSV denominado `cumpleanos_fechas.csv` contiene la lista de empleados con sus respectivas fechas de cumpleaños y direcciones de correo electrónico. La estructura del archivo tiene las siguientes columnas:

1. **Nombre:** El nombre del empleado.
2. **Fecha de Cumpleaños:** La fecha de cumpleaños en formato dd/mm/yyyy.
3. **Email:** La dirección de correo electrónico del empleado.

```
cumpleanos_fechas.csv > data
1  Nombre, Fecha de Cumpleaños, Email
2  TestNao, 02/11/2024, doc.hilariaadima.vasquez.du@unifranz.edu.bo
3  Inge Hadima, 18/12/2001, doc.hilariaadima.vasquez.du@unifranz.edu.bo
```

### 2.2 Obtención de la Fecha Actual

El script Batch ( `send_birthday_emails.sh` ) obtiene la fecha actual del sistema en formato `dd/mm/yyyy` . Esto se realiza utilizando el comando:

```
HOY=$(date +"%d/%m/%Y")
```

Esta fecha se utiliza para comparar con las fechas de cumpleaños almacenadas en el archivo CSV.

## 2.3 Iteración y Comparación

El script iterativamente lee cada línea del archivo CSV y compara la fecha de cumpleaños del empleado con la fecha de hoy. Esto se hace utilizando un bucle `while` que lee cada línea y verifica si la fecha coincide:

```
while IFS=',' read -r nombre fecha email
```

- Si la fecha coincide con la fecha de hoy, el script procede a enviar un correo.
- Si no hay coincidencia, se pasa al siguiente registro.

Se utiliza el comando `tail -n +2` para evitar la lectura de la cabecera del archivo CSV.

## 2.4 Envío del Correo Electrónico

En caso de que haya coincidencia en la fecha, el script imprime un mensaje en la consola indicando que se está enviando el correo y luego ejecuta un script de Python ( `emails_felicitaciones.py` ) que se encarga de enviar el correo. El script de Python recibe dos parámetros:

1. **Nombre del empleado:** Para personalizar el mensaje de felicitación.
2. **Correo electrónico:** Para enviar el correo al destinatario correcto.

El comando utilizado es:

```
python3 emails_felicitaciones.py "$nombre" "$email"
```

## 2.5 Seguridad y Manejo de Errores

El script está diseñado para evitar problemas comunes como el intento de lectura de una cabecera en el archivo CSV o envío de correos duplicados. También se omiten filas vacías u otros valores no deseados. El script Python está preparado para manejar errores de autenticación y otros posibles problemas durante el envío del correo.

```
#!/bin/bash

# Ruta del archivo CSV con la lista de cumpleaños
CSV_FILE="cumpleanos_fechas.csv"

# Obtener la fecha de hoy en formato dd/mm/yyyy
HOY=$(date +%d/%m/%Y)

# Iterar sobre cada línea del archivo CSV (exceptuando la cabecera)
while IFS=',' read -r nombre fecha email
do
    # Saltar la cabecera
    if [[ "$nombre" == "Nombre" ]]; then
        continue
    fi

    # Comprobar si la fecha coincide con la fecha actual
    if [[ "$fecha" == "$HOY" ]]; then
        echo "Enviando correo a $nombre ($email)..."
        # Ejecutar el script de Python para enviar el correo
        python3 emails_felicitaciones.py "$nombre" "$email"
    fi
done < <(tail -n +2 "$CSV_FILE")
```

## 2.6 Código del Script de Python

El script de Python `emails_felicitaciones.py` es responsable de enviar el correo electrónico de felicitación. A continuación se describe su funcionamiento:

1. **Configuración del Entorno:** Se utilizan variables de entorno para configurar la conexión SMTP, como el usuario y la contraseña del correo electrónico. Estas variables se cargan utilizando la librería `dotenv`.

```
from dotenv import load_dotenv
load_dotenv()
```

1. **Configuración del Servidor SMTP:** Se configura el servidor SMTP y el puerto para realizar el envío de correos mediante `smtplib`.

```
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT = 587
USERNAME = os.getenv('EMAIL_USERNAME')
PASSWORD = os.getenv('EMAIL_PASSWORD')
```

1. **Preparación del Mensaje:** Se personaliza el mensaje de felicitación utilizando un archivo HTML denominado `felicitaciones.html`. El contenido del archivo se lee y se reemplaza el marcador `{nombre_empleado}` con el nombre del destinatario.

```
with open('felicitaciones.html', 'r', encoding='utf-8') as file:
    cuerpo = file.read().replace("{nombre_empleado}", nombre_empleado)
```

1. **Envío del Correo:** Una vez configurado el mensaje, se realiza el envío del correo utilizando `smtplib`. Se manejan excepciones para errores de autenticación u otros errores durante el proceso de envío.

```
try:
    servidor = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
    servidor.starttls()
    servidor.login(USERNAME, PASSWORD)
    servidor.sendmail(USERNAME, destinatario, mensaje.as_string())
    servidor.quit()
    print(f'Correo enviado a {destinatario}')
except smtplib.SMTPAuthenticationError:
    print(f'Error de autenticación al enviar correo a {destinatario}: Verifique las credenciales.')
except Exception as e:
    print(f'Error al enviar correo a {destinatario}: {e}')
```

// CODIGO COMPLETO

```

import csv
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import os
from dotenv import load_dotenv
from datetime import datetime

# Cargar variables de entorno desde el archivo .env
load_dotenv()

# Configuración del servidor SMTP
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT = 587
USERNAME = os.getenv('EMAIL_USERNAME')
PASSWORD = os.getenv('EMAIL_PASSWORD') # Usar variable de
entorno para la contraseña

def enviar_saludo(destinatario, nombre_empleado):
    """
    Envía un saludo de cumpleaños al destinatario especificado.

    :param destinatario: Correo electrónico del destinatario.
    :param nombre_empleado: Nombre del empleado para personalizar el mensaje.
    """
    # Configurar el correo electrónico
    mensaje = MIMEMultipart()
    mensaje['From'] = USERNAME
    mensaje['To'] = destinatario
    mensaje['Subject'] = 'Saludos desde Soluciones innovadoras en el lenguaje de programación local'

    # Leer el cuerpo del correo desde el archivo HTML
    with open('felicitaciones.html', 'r', encoding='utf-8')

```

```

as file:
    cuerpo = file.read().replace("{nombre_empleado}", nombre_empleado)

    mensaje.attach(MIMEText(cuerpo, 'html'))

# Enviar el correo
try:
    servidor = smtplib.SMTP(SMTP_SERVER, SMTP_PORT)
    servidor.starttls()
    servidor.login(USERNAME, PASSWORD)
    servidor.sendmail(USERNAME, destinatario, mensaje.as_string())
    servidor.quit()
    print(f'Correo enviado a {destinatario}')
except smtplib.SMTPAuthenticationError:
    print(f'Error de autenticación al enviar correo a {destinatario}: Verifique las credenciales.')
except Exception as e:
    print(f'Error al enviar correo a {destinatario}: {e}')

def leer_lista_cumpleanos(archivo_csv):
    """
    Lee la lista de cumpleaños desde un archivo CSV y envía saludos si la fecha coincide con hoy.

    :param archivo_csv: Ruta del archivo CSV con la lista de cumpleaños.
    """
    hoy = datetime.now().strftime('%d/%m/%Y')
    with open(archivo_csv, mode='r') as file:
        reader = csv.reader(file)
        next(reader) # Saltar la cabecera
        for row in reader:
            if row and row[1] == hoy:
                nombre_empleado = row[0]

```

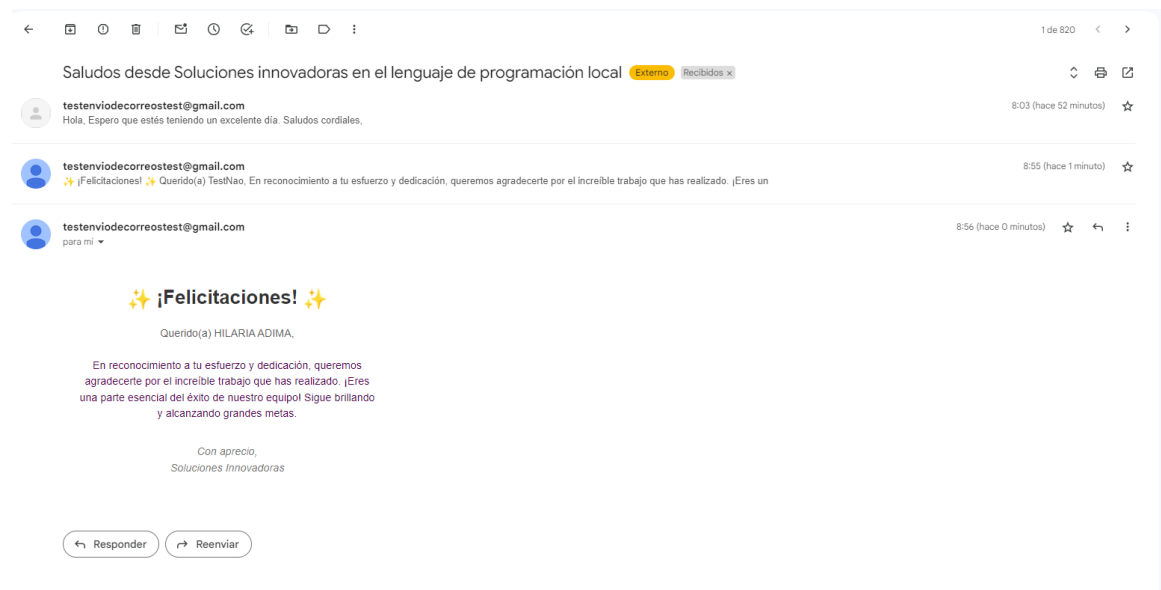
```
destinatario = row[2] # Utilizar el email
especificado en la tercera columna
enviar_saludo(destinatario, nombre_empleado)

# Ruta del archivo CSV con la lista de cumpleaños
archivo_csv = 'cumpleaños_fechas.csv'

# Ejecutar la función para leer la lista y enviar los saludos
leer_lista_cumpleaños(archivo_csv)
```

### 3. Resumen del Funcionamiento

1. **Preparación del Entorno:** Se definen variables necesarias como la ruta al archivo CSV y la fecha actual.
2. **Lectura del Archivo CSV:** Se itera sobre las filas, ignorando la cabecera.
3. **Comparación de Fechas:** Si la fecha de cumpleaños coincide con la fecha de hoy, se procede a enviar un correo.
4. **Envío del Correo:** Se ejecuta un script Python para realizar el envío del correo con un mensaje personalizado.
5. **Envío de Correo desde Python:** El script Python lee el contenido HTML, configura el mensaje y lo envía mediante SMTP.



## 4. Conclusiones

Este proceso permite automatizar el envío de felicitaciones de cumpleaños, lo cual contribuye a mejorar el ambiente laboral y la comunicación interna. La automatización reduce el riesgo de olvidos y facilita la gestión de los datos de los empleados. Asimismo, la implementación en Bash junto con Python permite una solución flexible y ampliable para futuras mejoras.

## 5. Mejoras Futuras

- **Integración con Bases de Datos:** Sustituir el archivo CSV por una base de datos para mejorar la seguridad y el acceso a los datos.
- **Notificaciones:** Implementar una confirmación automática para saber si el correo fue enviado exitosamente.
- **Interfaces de Usuario:** Añadir una interfaz gráfica para permitir que los usuarios gestionen la lista de cumpleaños y vean el estado de los envíos.