```
''' ENUNCIADO

    Una empresa posee un archivo maestro con los datos de sus empleados:
    DNI, Apellido y Nombre, y Sueldo. Una vez al mes se genera un archivo
    de Novedades con altas, bajas y modificaciones que deben ser aplicadas
    al archivo maestro.
    Ambos archivos están ordenados por DNI, y no caben totalmente en memoria.
    Escribir un programa modular en Python que obtenga un Maestro actualizado,
    teniendo en cuenta que en el archivos de novedades, vienen los mismos campos
    que en el archivo Maestro, más un campo en el cual puede venir una "A", para
    alta, una "B", para baja, y una "M" para modificación.
    Controlar que si es una modificación ó una baja, el DNI, debe existir en el
    maestro, sino enviar a un log de errores los datos (archivo txt), con la
    aclaracion del error.
    Si es un alta, no debe existir el DNI en el Maestro, de lo contrario enviar
    al Log de errores.
'''
#---------- Apareo de Archivos - Utilización del método standard -------------#

# Genera un maestro actualizado, en base al archivo maestro original y las
# novedades que vienen informadas en el archivo de novedades.
# Se controla que el tipo de operación a realizar Alta, Baja o Modificacion
# sea consistente con la logica de comparacion de claves.

# La funcion leerMaeNov utiliza el parametro "devolver", para poder ser
# utilizada tanto para cuando se debe leer el archivo Maestro y tener que
# devolver 3 vacios, como para también leer el archivo Novedades, y
# devolver 4 vacios cuando se encuentre con eof.

def leer_MaeNov(archivo, devolver):
    linea = archivo.readline()
    linea = linea.rstrip('\n')
    if not linea:
        linea = devolver
    return linea.split(",")

def grabar_MaeActualizado(archivo, legajo, nomApe, sueldo):
    archivo.write(legajo + ',' + nomApe + ',' + sueldo + '\n')

def grabar_error(archivo, legajo, nomApe, sueldo, tipo):
    archivo.write(legajo + ',' + nomApe + ',' + sueldo + ',' + tipo + '\n')

def aparearArchivos(arMaestro, arNovedades, arMaeActualizado, arLogErrores):

    # Tener en cuenta que en esta funcion estamos recibiendo los archivos
    # abiertos, si no estuvieramos seguros que nos encontramos al principio
    # de los archivos, antes de hacer la primer lectura, deberíamos aplicar
    # un seek(0), a arSuc1 y arSuc2, para asegurar que procesaremos los datos
    # desde el principio al final de cada archivo

    legajo_mae, nombre_mae, sueldo_mae = leer_MaeNov(arMaestro, ",,")
    legajo_nov, nombre_nov, sueldo_nov, tipo = leer_MaeNov(arNovedades, ",,,")

    while (legajo_mae and legajo_nov):
        if (legajo_mae < legajo_nov):
            # Va directo al nuevo archivo
            grabar_MaeActualizado(arMaeActualizado, legajo_mae, nombre_mae, sueldo_mae)
            # Vuelvo a leer Maestro
            legajo_mae, nombre_mae, sueldo_mae = leer_MaeNov(arMaestro, ",,")
```

```
        elif (legajo_nov < legajo_mae):
            # Deberia ser un alta
            if (tipo == 'A'): # si es alta la graba
                grabar_MaeActualizado(arMaeActualizado, legajo_nov, nombre_nov, sueldo_nov)
            else: # si no es alta graba en Log de Errores
                grabar_error(arLogErrores, legajo_nov, nombre_nov, sueldo_nov, "Baja o
                Modificacion de Legajo Inexistente")

            # Se vuelve a leer novedades
            legajo_nov, nombre_nov, sueldo_nov, tipo = leer_MaeNov(arNovedades, ",,,")

        else: # son iguales. Deberia ser una modificacion o una baja
            if (tipo == 'M'): # actualizo
                grabar_MaeActualizado(arMaeActualizado, legajo_nov, nombre_nov, sueldo_nov)
            elif (tipo == 'A'): # va al Log de Errores
                grabar_error(arLogErrores, legajo_nov, nombre_nov, sueldo_nov, "Alta de
                Legajo Existente")
                # En este caso, el que estaba en el Maestro, lo dejo igual
                grabar_MaeActualizado(arMaeActualizado,legajo_mae, nombre_mae, sueldo_mae)
                # Si fuese una "B" (baja), no hacemos nada

            # Leo los dos
            legajo_mae, nombre_mae, sueldo_mae = leer_MaeNov(arMaestro,",,")
            legajo_nov, nombre_nov, sueldo_nov, tipo = leer_MaeNov(arNovedades,",,,")

    # Del while puedo estar saliendo porque encontre el final de ambos archivos
    # o porque encontre el de solo uno. Por lo tanto, puede haber registros a
    # a procesar y debo hacerlo

    while legajo_mae:
        grabar_MaeActualizado(arMaeActualizado,legajo_mae, nombre_mae, sueldo_mae)
        legajo_mae, nombre_mae, sueldo_mae = leerMaeNov(arMaestro,",,")

    while legajo_nov:
        if (tipo == "A"):
            grabar_MaeActualizado(arMaeActualizado, legajo_nov, nombre_nov, sueldo_nov)
        else:
            grabar_error(arLogErrores, legajo_nov, nombre_nov, sueldo_nov, tipo)
        legajo_nov, nombre_nov, sueldo_nov, tipo = leerMaeNov(arNovedades,",,,")


#---------------------------- BLOQUE PRINCIPAL ------------------------------------#

arMaestro = open("empleados_maestro.csv","r")
arNovedades = open("empleados_novedades.csv","r")
arMaeActualizado = open('empleados_maestro_actual.csv','w')
arLogErrores = open('empleados_logErrores.txt','w')

aparearArchivos(arMaestro, arNovedades, arMaeActualizado, arLogErrores)

arMaestro.close()
arNovedades.close()
arMaeActualizado.close()
arLogErrores.close()
```