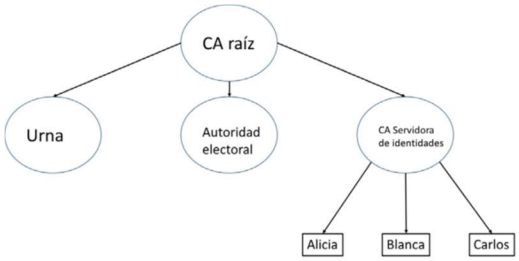
PRÁCTICA 5. FIRMA Y VERIFICACIÓN RSA

Para la realización de la práctica 5 se debe tener en cuenta el diagrama de voto de la práctica 3 que se muestra debajo. En este caso, se ha realizado la práctica suponiendo que Alicia envía su voto a la urna, por lo tanto, necesitaremos tanto el certificado como la clave privada de Alicia generada en la práctica anterior. Estos archivos son necesarios dado que hay que comprobar que nadie ha suplantado la identidad de el votante que en este caso será Alicia. Ya que de ser así la función de verificación deberá devolver un FALSE indicando así que o bien en la firma o bien en la verificación se ha suplantado la identidad.

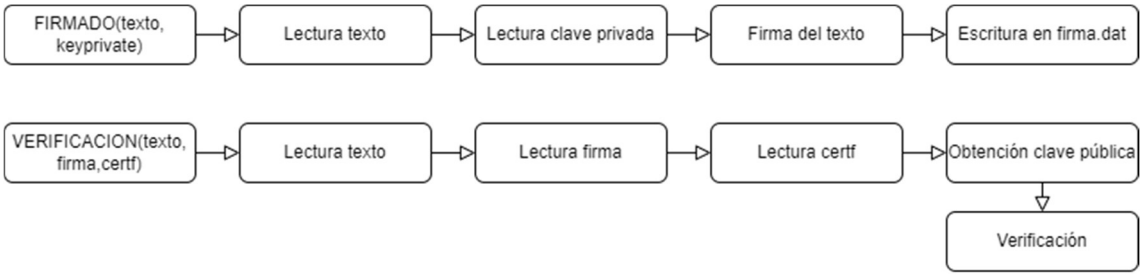


Las funciones de firma y verificación RSA se pueden realizar de dos maneras, o bien obteniendo tanto los números públicos como privados de las claves para cada una de las acciones o bien mediante las funciones sign y verify con los parámetros de entrada necesarios. En el caso de las funciones sign y verify el procedimiento sería:

```
signature = private_key.sign(message,
    padding.PSS(
      mgf=padding.MGF1(hashes.SHA256()),
      salt_length=padding.PSS.MAX_LENGTH
    ),
    hashes.SHA256()
  )
```

```
keypublic.verify( signature, message,
      padding.PSS(
        mgf=padding.MGF1(hashes.SHA256()),
        salt_length=padding.PSS.MAX_LENGTH
      ),
      hashes.SHA256()
    )
```

En mi caso, el método utilizado ha sido el segundo dado que el primero conllevaba mucho tiempo de ejecución. Para la definición de las funciones de firma y verificación he seguido los siguientes diagramas:



Es preciso mencionar que para que las funciones de firma y verificación funcionen correctamente tanto con clave pública como privada los ficheros se leen y escriben mediante “rb” y “wb”, es decir, en bytes.

Teniendo todo esto en cuenta el programa principal quedará de la siguiente manera:

```
def main():
    key = b"HOLA"
    with open("resumen.dat", "wb") as key_file:
        key_file.write(key)
    key_file.close()

    print(firmado("resumen.dat","Alicia.pem"))
    print(verificacion("resumen.dat","firma.dat","Alicia.crt"))

if __name__ == '__main__':
        main()
```

Por lo tanto, para ejecutar el programa únicamente son necesarios los ficheros “resumen.dat”, “Alicia.crt” y “Alicia.pem” en el mismo directorio que el fichero fuente y una vez ejecutado se creará el fichero “firma.dat.dat” y devolverá por pantalla un True o False dependiendo si la verificación es correcta o no. Cabe mencionar que el fichero original de la práctica contiene el hash del voto generado en la práctica 2.