```
app = Flask(__name__)
csrf = CSRFProtect()
csrf.init_app(app) # Compliant
m = sha.new()
```

a web application can be forced, by an attacker, to perform sensitive actions that he didn't
anything that can change the state of the application.

e privileged action, or to visit a malicious web site that embeds a hidden web request and
ticated and sensitive.

```
utf-8"))
de.hexdigest()
```

Práctica 4: Jorge Andrés Mármol Rivera

Hemos elegido el proyecto de la agenda. Tras analizarlo encontramos el error:

```
def md5_encode(self,text):
    encode = hashlib.md5()
    encode.update(text.encode("utf-8"))
    return encode.hexdigest()
```

Cryptographic hash algorithms such as `MD2`, `MD4`, `MD5`, `MD6`, `HAVAL-128`, `HMAC-MD5`, `DSA` (which uses `SHA-1`), `RIPEMD`, `RIPEMD-128`, `RIPEMD-160`, `HMACRIPEMD160` and `SHA-1` are no longer considered secure, because it is possible to have `collisions` (little computational effort is enough to find two or more different inputs that produce the same hash).

El error que marca es que este tipo de criptografía puede no ser seguro. Parece que la librería ha dejado de ser actualizada, ya que como se puede observar para arreglar el error:

```
import md5 //
m = md5.new()

import sha //
m = sha.new()
```

Una de las formas es importar otra librería e invocar una función de ella misma.

Otra forma que se indica es con:

```
import hashlib
m = hashlib.sha512() // Compliant
```

Además, las vulnerabilidades marcadas son :

- OWASP Top 10 2017 Category A3 - Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 - Security Misconfiguration

Ambas pueden ser causadas por la criptografía, el de tipo A3 provoca un error de exposición de datos y el de A6 de configuración incorrecta.