

Generar hash SHA y encriptar cadenas en JAVA

Muchas veces es necesario cifrar, encriptar o generar hashes de passwords u otros textos delicados. Para realizar esto en Java se utiliza la clase MessageDigest la cual nos permite cargar una algoritmo y encriptarlo en forma de bytes.

La clase MessageDigest permite el uso de los algoritmos MD2, MD5, SHA-1, SHA-256, SHA-384, SHA-512.

En el siguiente código se puede observar un ejemplo de MD5, SHA-1 y SHA-512:

```
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import org.apache.commons.codec.binary.Hex;

public class DigestExample {

    public static void main(String[] args) {

        MessageDigest md = null;
        String password = "password D:";
        try {

            //SHA-512
            md= MessageDigest.getInstance("SHA-512");
            md.update(password.getBytes());
            byte[] mb = md.digest();
            System.out.println(Hex.encodeHex(mb));

            //SHA-1
            md= MessageDigest.getInstance("SHA-1");
            md.update(password.getBytes());
            mb = md.digest();
            System.out.println(Hex.encodeHex(mb));
```

```

        //MD5
        md= MessageDigest.getInstance("MD5");
        md.update(password.getBytes());
        mb = md.digest();
        System.out.println(Hex.encodeHex(mb));

    } catch (NoSuchAlgorithmException e) {
        //Error
    }
}
}

```

[view rawDigetsExample.java](#) hosted with ❤ by [GitHub](#)

Como se puede observar simplemente obtenemos la instancia del algoritmo que se desea, le agregamos la cadena en forma de bytes con el método update y lo transformamos. Para poder visualizarlo en forma Hexadecimal se utiliza la clase Hex que se obtiene de la librería commons-codec de Apache, la cual se puede descargar en el siguiente [link](#) o directamente de la pagina de Apache.

Al final se obtiene lo siguiente:

```

793d5ecd344bd91c808917d8821ae74b4c84550964b67906bdeb6c7dc0146bef90b5a2
b326ac31049b0e3930a0d697b760c13236596da2d142e6947a1e3386b5
8e8725edeaa090c62110e9c7b23a243873d61b36
1462a2da3e9d67e5824c71b2ac91bc66

```