

# API Documentation

API Documentation

April 12, 2015

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Module cifrados</b>	<b>2</b>
1.1 Variables . . . . .	2
1.2 Class Cesar . . . . .	2
1.2.1 Methods . . . . .	2
1.3 Class Afin . . . . .	3
1.3.1 Methods . . . . .	3
1.4 Class Mochila . . . . .	4
1.4.1 Methods . . . . .	4
1.5 Class RSA . . . . .	6
1.5.1 Methods . . . . .	6
<b>2 Module funciones</b>	<b>8</b>
2.1 Functions . . . . .	8
2.2 Variables . . . . .	12
<b>Index</b>	<b>13</b>

# 1 Module cifrados

Modulo con las clases de cifrado

## 1.1 Variables

Name	Description
<code>__package__</code>	<b>Value:</b> None

## 1.2 Class Cesar

Cifrado Cesar

### 1.2.1 Methods

<b><code>__init__</code></b> ( <i>self</i> )
Constructor de clase

<b><code>setK</code></b> ( <i>self</i> , <i>k</i> )
Modificar de k
<b>Parameters</b>
<i>k</i> : valor de la clave
( <i>type</i> = <i>entero</i> )

<b><code>setTexto</code></b> ( <i>self</i> , <i>texto</i> )
Modificador del texto
<b>Parameters</b>
<i>texto</i> : texto a cifrar
( <i>type</i> = <i>cadena</i> )

<b><code>cifrar</code></b> ( <i>self</i> )
Cifrado del texto por el método César
<b>Return Value</b>
texto cifrado
( <i>type</i> = <i>cadena</i> )

**descifrar**(*self*)

Descifrado del texto por el método César

**Return Value**

texto descifrado

*(type=cadena)*

## 1.3 Class Afin

Cifrado Afín

### 1.3.1 Methods

**\_\_init\_\_**(*self*)

Constructor de clase

**setA**(*self*, *a*)

Modificador de la clave a

**Parameters****a**: valor de a*(type=entero)***setD**(*self*, *d*)

Modificador de la clave d

**Parameters****d**: valor de d*(type=entero)***setTexto**(*self*, *texto*)

Modificar del texto

**Parameters****texto**: cadena a cifrar*(type=cadena)***cifrar**(*self*)

Cifrado del texto por el método afín

**Return Value**

texto cifrado

*(type=cadena)*

---

**descifrar**(*self*)
 

---

 Descifrado del texto por el método afin
 

---

**Return Value**

 texto descifrado
 

---

*(type=cadena)*

## 1.4 Class Mochila

Cifrado mochila o de Merkle-Hellman

### 1.4.1 Methods

---

**\_\_init\_\_**(*self*)
 

---

 Constructor de la clase
 

---



---

**setMochila**(*self*, *mochila*)
 

---

 Modificador de la clave mochila
 

---

**Parameters**

 mochila: valor de la clave mochila
 

---

*(type=lista)*


---

**setM**(*self*, *m*)
 

---

 Modificador de la clave m
 

---

**Parameters**

 m: valor de la clave m
 

---

*(type=entero)*


---

**setW**(*self*, *w*)
 

---

 Modificador de la clave w
 

---

**Parameters**

 w: valor de la clave w
 

---

*(type=entero)*


---

**setPublic**(*self*, *public*)
 

---

 Modificador de la clave publica
 

---

**Parameters**

 public: valor de la lista publica
 

---

*(type=lista)*

**setTexto**(*self*, *texto*)

Modificador del texto a cifrar/descifrar

**Parameters**

**texto:** texto a cifrar/descifrar  
(*type=cadena*)

**setClaves**(*self*, *mochila*, *m*, *w*, *public*)

Modificador de todas las claves

**Parameters**

**mochila:** cadena con los valores de la mochila  
(*type=cadena*)

**m:** valor de la clave m  
(*type=entero*)

**w:** valor de la clave w  
(*type=entero*)

**public:** cadena con los valroes de la clave publica  
(*type=cadena*)

**cifrar**(*self*)

Cifrado por el método mochila

**Return Value**

lista con los valores del cifrado  
(*type=lista*)

**descifrar**(*self*)

Descifrado por el método mochila

**Return Value**

texto descifrado  
(*type=cadena*)

**generarClave**(*self*)

Genera las claves aleatoriamente y las modifica en la clase

**descifraNumero**(*self*, *texto*, *mochila*)

Consigue la cadena de binarios para descifrar

**Return Value**

cadena con los numeros binarios para descifrar el texto  
(*type=cadena*)

## 1.5 Class RSA

Cifrado RSA

### 1.5.1 Methods

<b><code>__init__</code></b> ( <i>self</i> )
Constructor de clase
<b><code>setN</code></b> ( <i>self</i> , <i>n</i> )
Modificador de la clave n
<b>Parameters</b> <i>n</i> : valor de la clave n <i>(type=entero)</i>
<b><code>setE</code></b> ( <i>self</i> , <i>e</i> )
Modificador de la clave e
<b>Parameters</b> <i>e</i> : valor de la clave e <i>(type=entero)</i>
<b><code>setD</code></b> ( <i>self</i> , <i>d</i> )
Modificador de la clave d
<b>Parameters</b> <i>d</i> : valor de la clave d <i>(type=entero)</i>
<b><code>setTexto</code></b> ( <i>self</i> , <i>texto</i> )
Modificador del texto a cifrar
<b>Parameters</b> <i>texto</i> : texto a descifrar <i>(type=cadena)</i>

**cifrar**(*self*, *e*, *n*, *texto*)

Cifrado por el método RSA

**Parameters**

**e:** valor de e  
(*type=entero*)

**n:** valor de la clave n  
(*type=entero*)

**texto:** texto a cifrar  
(*type=cadena*)

**Return Value**

valores del texto cifrado  
(*type=lista*)

**descifrar**(*self*, *d*, *n*, *cifrado*)

Descifrado por el método RSA

**Parameters**

**d:** clave d  
(*type=entero*)

**n:** clave n  
(*type=entero*)

**cifrado:** lista con los valores a descifrar  
(*type=lista*)

**Return Value**

texto descifrado  
(*type=cadena*)

**generarFirma**(*self*, *texto*)

Genera una firma digital por RSA

**Parameters**

**texto:** texto con el que generar la firma  
(*type=cadena*)

**Return Value**

valor de la firma digital  
(*type=entero*)

**generarClave**(*self*)

Generacion de claves para RSA

## 2 Module funciones

Modulo con las funciones generales

### 2.1 Functions

#### **letranumero**(*texto*)

Convierte una letra a su numero correposdiente

##### **Parameters**

**texto:** cadena a convertir a numeros  
(*type=cadena*)

##### **Return Value**

lista de numeros  
(*type=lista*)

#### **numeroLetra**(*num*)

Convierte un numero a su letra correspondiente

##### **Parameters**

**num:** lista de numeros a convertir  
(*type=lista*)

##### **Return Value**

cadena de los numeros convertidos a letras  
(*type=cadena*)

#### **abecedario**()

Alfabeto a usar

##### **Return Value**

alfabeto a usar  
(*type=cadena*)

#### **egcd**(*a, b*)

MCD de dos numeros

##### **Parameters**

**a:** primer elemento para calcular el mcd  
(*type=entero*)  
**b:** segundo elemento para calcular el mcd  
(*type=entero*)

##### **Return Value**

Maximo común divisor y los coeficientes de Bézout  
(*type=lista*)



**modinv**(*a*, *m*)

Calcula el inverso de un numero modulo m

**Parameters**

**a:** numero a calcular el inverso

(*type=entero*)

**m:** modulo en el que calcular el inverso

(*type=entero*)

**Return Value**

inverso de a modulo m

(*type=entero*)

**numeroBinario**(*texto*)

Convierte una cadena en su respectivos numeros binarios de 5 bits

**Parameters**

**texto:** texto a convertir

(*type=cadena*)

**Return Value**

cadena de numeros de 5 bits

(*type=cadena*)

**binarioNumero**(*texto*)

Convierte una cadena de numeros binarios de 5 bits a numeros decimales

**Parameters**

**texto:** cadena de numeros binarios de 5 bits

(*type=cadena*)

**Return Value**

cadena numeros decimales

(*type=lista*)

**generarPrimos**(*n*)

Genera un primo aleatorio

**Parameters**

**n:** escoge entre numeros primos de 0 a 5000 o de 0 a 150

(*type=entero*)

**Return Value**

numero primo

(*type=entero*)

**listacadena**(*lista*)

Convierte una lista en cadena

**Parameters**

**lista:** lista a convertir  
(*type=lista*)

**Return Value**

lista convertida a cadena  
(*type=cadena*)

**cadenaLista**(*cadena*)

Convierte una cadena de numeros separados por ',' en lista

**Parameters**

**cadena:** cadena a convertir  
(*type=cadena*)

**Return Value**

cadena convertida a lista  
(*type=lista*)

**letraNumero2D**(*texto*)

Convierte la cadena en una cadena de numeros de dos cifras

**Parameters**

**texto:** cadena a convertir en numeros  
(*type=cadena*)

**Return Value**

cadena de numeros de dos cifras  
(*type=cadena*)

**dec2bin**(*a*)

Convierte un entero a binario

**Parameters**

**a:** numero a convertir  
(*type=entero*)

**Return Value**

numero binario  
(*type=cadena*)

**potencia**(*c, d, n*)

Potencia modulo n

**Parameters****c:** numero a elevar*(type=entero)***d:** potencia a la que elevar c*(type=entero)***n:** modulo*(type=entero)***Return Value**

c elevado a d modulo n

*(type=entero)***reshape**(*cadena, n*)

Divide una cadena en una lista de elementos de longitud n

**Parameters****cadena:** cadena a dividir*(type=cadena)***n:** longitud de los elementos de la lista*(type=entero)***Return Value**

lista con la cadena dividida

*(type=lista)***prepa\_num\_cifrar**(*n, texto*)

Prepara el texto para poder cifrarlo por RSA

**Parameters****n:** clave n*(type=entero)***texto:** texto a cifrar*(type=cadena)***Return Value**

lista con los numeros preparados para cifrar

*(type=lista)*

<b>num_letra</b> ( <i>n</i> , <i>cifrado</i> )
descifra el vector para descifrarlo por RSA
<b>Parameters</b>
<b>n</b> : clave n ( <i>type=entero</i> )
<b>cifrado</b> : lista de numeros a descifrar ( <i>type=lista</i> )
<b>Return Value</b>
texto descifrado ( <i>type=cadena</i> )

<b>hash</b> ( <i>s</i> , <i>M</i> )
Funcion Hash
<b>Parameters</b>
<b>s</b> : cadena con la que realizar el hash ( <i>type=cadena</i> )
<b>M</b> : modulo de la funcion ( <i>type=entero</i> )
<b>Return Value</b>
resumen del mensaje ( <i>type=entero</i> )

## 2.2 Variables

Name	Description
__package__	<b>Value:</b> None

## Index

- cifrados (*module*), 2–7
  - cifrados.Afin (*class*), 3–4
    - cifrados.Afin.\_\_init\_\_ (*method*), 3
    - cifrados.Afin.cifrar (*method*), 3
    - cifrados.Afin.descifrar (*method*), 3
    - cifrados.Afin.setA (*method*), 3
    - cifrados.Afin.setD (*method*), 3
    - cifrados.Afin.setTexto (*method*), 3
  - cifrados.Cesar (*class*), 2–3
    - cifrados.Cesar.\_\_init\_\_ (*method*), 2
    - cifrados.Cesar.cifrar (*method*), 2
    - cifrados.Cesar.descifrar (*method*), 2
    - cifrados.Cesar.setK (*method*), 2
    - cifrados.Cesar.setTexto (*method*), 2
  - cifrados.Mochila (*class*), 4–5
    - cifrados.Mochila.\_\_init\_\_ (*method*), 4
    - cifrados.Mochila.cifrar (*method*), 5
    - cifrados.Mochila.descifraNumero (*method*), 5
    - cifrados.Mochila.descifrar (*method*), 5
    - cifrados.Mochila.generarClave (*method*), 5
    - cifrados.Mochila.setClaves (*method*), 5
    - cifrados.Mochila.setM (*method*), 4
    - cifrados.Mochila.setMochila (*method*), 4
    - cifrados.Mochila.setPublic (*method*), 4
    - cifrados.Mochila.setTexto (*method*), 4
    - cifrados.Mochila.setW (*method*), 4
  - cifrados.RSA (*class*), 5–7
    - cifrados.RSA.\_\_init\_\_ (*method*), 6
    - cifrados.RSA.cifrar (*method*), 6
    - cifrados.RSA.descifrar (*method*), 7
    - cifrados.RSA.generarClave (*method*), 7
    - cifrados.RSA.generarFirma (*method*), 7
    - cifrados.RSA.setD (*method*), 6
    - cifrados.RSA.setE (*method*), 6
    - cifrados.RSA.setN (*method*), 6
    - cifrados.RSA.setTexto (*method*), 6
- funciones (*module*), 8–12
  - funciones.abecedario (*function*), 8
  - funciones.binarioNumero (*function*), 9
  - funciones.cadenaLista (*function*), 10
  - funciones.dec2bin (*function*), 10
  - funciones.egcd (*function*), 8
  - funciones.generarPrimos (*function*), 9
  - funciones.hash (*function*), 12
  - funciones.letranumero (*function*), 8
  - funciones.letraNumero2D (*function*), 10
  - funciones.listacadena (*function*), 9
  - funciones.modinv (*function*), 8
  - funciones.num\_letra (*function*), 11
  - funciones.numeroBinario (*function*), 9
  - funciones.numeroLetra (*function*), 8
  - funciones.potencia (*function*), 10
  - funciones.prepa\_num\_cifrar (*function*), 11
  - funciones.reshape (*function*), 11