PRACTICA 5 SPSI
Puzles Hash

Antonio Manuel Rodriguez Martos

Todo el código será realizado en python. El código estará disponible en formato texto al final del documento.

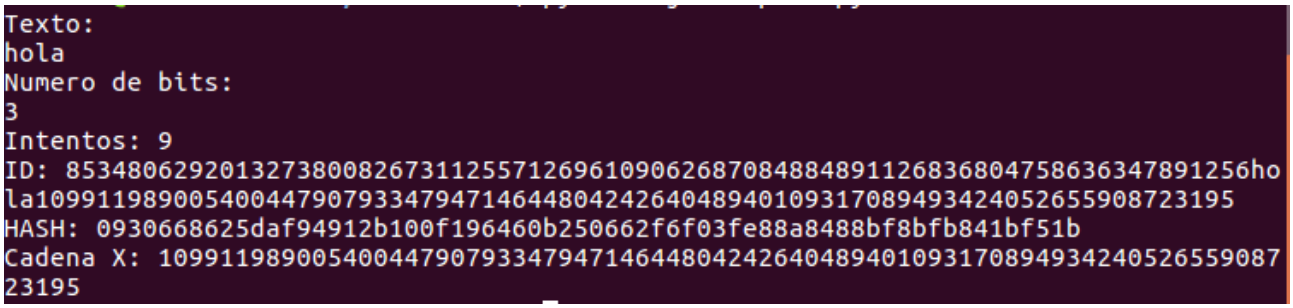
1.)

En este primer ejercicio generaremos una cadena aleatoria llamada ‘cad’ a la cual le asignaremos el string ‘texto’ que pasaremos por la terminal y se lo asignaremos a la variable id.
Después haremos la busqueda con un bucle que no terminará hasta encontrar el número de ceros ‘bits’ que le hayamos pasado por terminal. Dentro del mismo se generará una nueva cadena aleatoria la cual juntaremos con el ‘id’ anterior, será el ‘nuevo_id’.
La función hash con n >=256 que hemos escogido para realizar esta práctica será sha256.
Calcularemos el hash al nuevo id y buscaremos si coinciden en binario el número de ceros pedidos.

```
3
4   import hashlib
5   import random
6
7   num = 256
8
9   print "Texto: "
10  texto = raw_input()
11
12  print "Numero de bits: "
13  bits = int(raw_input())
14
15  #Genaramos cadena aleatoria de num bits
16  cad = random.getrandbits(num)
17  #Concatenamos cadena con texto
18  id = str(cad) + texto
19
20  encontrado = False
21  intentos = 0
22  while not encontrado:
23      intentos += 1
24
25      #Generamos nueva cadena aleatoria de num bits
26      nueva_cad = random.getrandbits(num)
27      #Concatenamos el id con la nueva cadena
28      nuevo_id = id + str(nueva_cad)
29
30      #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
31      hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
32      #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
33      cad_bin = bin(int(hash, 16))[2:].rjust(num, '0')
34
35      #Buscamos el numero de ceros bits
36      cont = 0
37      for i in range(0,bits):
38          if (cad_bin[i] == "0"):
39              cont += 1
40      #Si el numero de ceros coinciden con el numero de introducido
41      if (cont == bits):
42          encontrado = True
43          print "Intentos:", intentos
44          print "ID:", nuevo_id
45          print "HASH:", hash
46          print "Cadena X:", nueva_cad
47
```

Código apartado 1

Un ejemplo de ejecutar el código anterior lo podemos ver en la siguiente imagen.



Ejemplo resultado ejecucion código ap.1

Como vemos en este caso buscamos 3 ceros y comprobamos que efectivamente el hash dado los tiene.

2.)

Modificaremos el código del apartado 1 definiendo la función ‘buscarCeros’ a la que le pasaremos el texto , el numero de bits y una cadena. En este caso al primer bloque le pasaremos una cadena aleatoria, a los demás le pasaremos el hash del anterior.

```
18  def buscarCeros(texto, bits, cadena):
19
20      #Concatenamos cadena con texto
21      id = str(cadena) + texto
22
23      encontrado = False
24      intentos = 0
25      while not encontrado:
26          intentos += 1
27
28          #Generamos nueva cadena aleatoria de num bits
29          nueva_cad = random.getrandbits(num)
30          #Concatenamos el id con la nueva cadena
31          nuevo_id = id + str(nueva_cad)
32
33          #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
34          hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
35          #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
36          cad_bin = bin(int(hash, 16))[2:].rjust(256, '0')
37
38          #Buscamos el numero de ceros bits
39          #Si no coinciden sumo 1 a la nueva cadena
40          cont = 0
41          for i in range(0,bits):
42              if (cad_bin[i] == "0"):
43                  cont += 1
44
45          if (cont == bits):
46              encontrado = True
47              print "Intentos:", intentos
48              print "ID:", nuevo_id
49              print "HASH:", hash
50              print "Cadena X:", nueva_cad
51
52      return hash
53
54  print "-------------------------------------------------------------------------"
55  print "Bloque 0"
56  hash = buscarCeros(text, b, cad)
57  print "-------------------------------------------------------------------------"
58  for i in range(9):
59      print "Bloque",i+1
60      hash = buscarCeros(text, b, hash)
61      print "-------------------------------------------------------------------------"
62
```

Código apartado 2

Un resultado de ejecutar el código anterior :

Bloque 0
Intentos: 1
ID:
995978399022767429032479012839011139748000997186333002656752472211435845913 4antoniomanuelrodriguez
martos10231173874329095251962006982308641033709109441701505282825651461196997022968 6
HASH: 04cb187d76805bd60f608b6a401659dd7a4be0276078fc0247f77e4249772a99
Cadena X: 10231173874329095251962006982308641033709109441701505282825651461196997022968 6

Bloque 1
Intentos: 2
ID:
04cb187d76805bd60f608b6a401659dd7a4be0276078fc0247f77e4249772a99antoniomanuelrodriguezmartos764750550
0544708938561282869208303397639565918661964817226243281557946764854
HASH: 0bc91f9a8c2d15f8e20ffbf3cba87e3337e449b5201d1d24c9a6fbc1207225ec
Cadena X: 7647505500544708938561282869208303397639565918661964817226243281557946764854

Bloque 2
Intentos: 1
ID:
0bc91f9a8c2d15f8e20ffbf3cba87e3337e449b5201d1d24c9a6fbc1207225ecantoniomanuelrodriguezmartos9966545700
9756255529261269133139158227243996104557745793098839577365303482473
HASH: 38d0bda635b312a1a0ec1c2f8a7c19963c26adf6f8da12357ceafc36f980b5da
Cadena X: 9966545700975625552926126913313915822724399610455774579309883957736530348247 3

Bloque 3
Intentos: 1
ID:
38d0bda635b312a1a0ec1c2f8a7c19963c26adf6f8da12357ceafc36f980b5daantoniomanuelrodriguezmartos5868644501
0907616267461792438346677247167593612545232671539804070053992029777
HASH: 37bbafe8efe6f8c7c5fd8c94b120ce541fda5d65e8b5897024a3f2c22475b3ac
Cadena X: 5868644501090761626746179243834667724716759361254523267153980407005399202977 7

Bloque 4
Intentos: 4
ID:
37bbafe8efe6f8c7c5fd8c94b120ce541fda5d65e8b5897024a3f2c22475b3acantoniomanuelrodriguezmartos38247109028
3307552261666385972815501691359097762013950989504965519245368883 77
HASH: 3cf8b8a47ed5f8922737676b5040c8d835061e9efd2fccf86f63e0782f3dfa2b
Cadena X: 3824710902833075522616663859728155016913590977620139509895049655192453688837 7

Bloque 5
Intentos: 2
ID:
3cf8b8a47ed5f8922737676b5040c8d835061e9efd2fccf86f63e0782f3dfa2bantoniomanuelrodriguezmartos60800493778
9385616007975422300601063694436835939961693222801449539579881271 37
HASH: 1d7805b4362abe5ebf48f4ce7ad378c9f78955ddf78405b46d5f770a0fc55c22
Cadena X: 6080049377893856160079754223006010636944368359399616932228014495395798812713 7

Bloque 6
Intentos: 1
ID:
1d7805b4362abe5ebf48f4ce7ad378c9f78955ddf78405b46d5f770a0fc55c22antoniomanuelrodriguezmartos5694830467
2161637338963621443709009977265483415199414075234115559966390513111
HASH: 1f527243f2a749c91906e65cfe02fd614386b07ffecbb799450265f3655a2a75
Cadena X: 5694830467216163733896362144370900997726548341519941407523411555996639051311 1

Bloque 7
Intentos: 7
ID:
1f527243f2a749c91906e65cfe02fd614386b07ffecbb799450265f3655a2a75antoniomanuelrodriguezmartos6074528760
0934311410313742191652151449641097792302657285922490538237192722179
HASH: 1993c1c323de69b3eee381119dec816b29473184da24eebc473931a658e8086c

Cadena X: 607452876009343114103137421916521514496410977923026572859224905382371927221 79

Bloque 8
Intentos: 5
ID:
1993c1c323de69b3eee381119dec816b29473184da24eebc473931a658e8086cantoniomanuelrodriguezmartos404366589
657438512607303394843275146553945843523807988015351444880288797816 27
HASH: 198bab1da69c8bbbf5a4e8e297b76e2d1c3e638111bfa16e5bfd826d430b43e3
Cadena X: 40436658965743851260730339484327514655394584352380798801535144488028879781627

Bloque 9
Intentos: 6
ID:
198bab1da69c8bbbf5a4e8e297b76e2d1c3e638111bfa16e5bfd826d430b43e3antoniomanuelrodriguezmartos216294330
591374212632923671479740917897746142766828048528051161023363728188 34
HASH: 36e2e0cd96e27159dc5d15f8afa2fbbfc3f06398fe1923224774e3eac4093cf9
Cadena X: 2162943305913742126329236714797409178977461427668280485280511610233637281883 4

3.)

Añadiremos unas lineas de código adicionales al código del apartado anterior.

```
65  b= 3
66
67  for i in range(10):
68      print "Bloque",i+10
69      hash = buscarCeros(text, b, hash)
70      print "------------------------------------------------------------------------"
71
```

Parte del código apartado 3

Un resultado con los primeros diez bloques con b = 2 y los diez siguientes con b = 3 :

Bloque 0
Intentos: 9
ID:
4980026069407836847740335929167492294587339758434240425830270735604791137569 5antoniomanuelrodrigue
zmartos998800636954038669015514341696329580418003716831552236671437407226524726033 7
HASH: 070fb02f9a9b36af09885f89925b7989c1a90722a9f74585e3e5ffc8f106083c
Cadena X: 998800636954038669015514341696329580418003716831552236671437407226524726033 7

Bloque 1
Intentos: 4
ID:
070fb02f9a9b36af09885f89925b7989c1a90722a9f74585e3e5ffc8f106083cantoniomanuelrodriguezmartos75928288792
903480554250854469825854426840992222094964466922329932827100690571
HASH: 09a9f87e135af60117e3cc7a9f19df066b2551a02ccb4d943f15ac1854a0aced
Cadena X: 7592828879290348055425085446982585442684099222094964466922329932827100690571

Bloque 2
Intentos: 2
ID:
09a9f87e135af60117e3cc7a9f19df066b2551a02ccb4d943f15ac1854a0acedantoniomanuelrodriguezmartos9532835219
0098640507987465356332620553729550017216840769130136587541246347124
HASH: 3e0058b11b69fb7df254db017a8d9af0876f832bbee38eb073061c9bc46c7226
Cadena X: 95328352190098640507987465356332620553729550017216840769130136587541246347124

Bloque 3
Intentos: 2
ID:
3e0058b11b69fb7df254db017a8d9af0876f832bbee38eb073061c9bc46c7226antoniomanuelrodriguezmartos295399775
69369148908767807526155275039342012648667855082110764116973656778341
HASH: 08a17e9e367ad98a5393cbd07cc96eb5b629bc3c0c7077105671264a68bbdec2
Cadena X: 29539977569369148908767807526155275039342012648667855082110764116973656778341

Bloque 4
Intentos: 1
ID:
08a17e9e367ad98a5393cbd07cc96eb5b629bc3c0c7077105671264a68bbdec2antoniomanuelrodriguezmartos111773171
08456189295571873737478758039172836501370749650324120095441945578807 8
HASH: 24211d86dd58d22a8f6d0556995286626ba272bc22485ae1ce8b2786b1526f45
Cadena X: 11177317108456189295571873737478758039172836501370749650324120095441945578807 8

Bloque 5
Intentos: 1
ID:
24211d86dd58d22a8f6d0556995286626ba272bc22485ae1ce8b2786b1526f45antoniomanuelrodriguezmartos95000766
76788038507827032872308500182546643710857064757686980439676123602824 1
HASH: 3feadde44068219065829a9d3e0bed04f04808cf29a54e1ed958e78fa96ddd95
Cadena X: 95000766767880385078270328723085001825466437108570647576869804396761236028241

Bloque 6
Intentos: 2
ID:
3feadde44068219065829a9d3e0bed04f04808cf29a54e1ed958e78fa96ddd95antoniomanuelrodriguezmartos1101893883
32494095875166498621485515862835758035573158601104566875019249949513
HASH: 056f17d9098807a7257b31f0cb0cfd26b259559bf6eb14f365728b16f6bbef9b
Cadena X: 11018938833249409587516649862148551586283575803557315860110456687501924994951 3

Bloque 7
Intentos: 4
ID:
056f17d9098807a7257b31f0cb0cfd26b259559bf6eb14f365728b16f6bbef9bantoniomanuelrodriguezmartos8995275840
35982961289394716314204556704505126203832128392013040247747304524 0
HASH: 290a26a81636ad67491b142ab95147d023e77c689057b28472ea7d25c5b4f3d9
Cadena X: 89952758403598296128939471631420455670450512620383212839201304024774730452420

Bloque 8
Intentos: 3
ID:
290a26a81636ad67491b142ab95147d023e77c689057b28472ea7d25c5b4f3d9antoniomanuelrodriguezmartos25375971
13917001110971544573038004197748015665131043215044912544131647897786 3
HASH: 186033dbf595faf33c27e84b984940a90dba668f9c81806aaafed504f5c04eee
Cadena X: 25375971139170011109715445730380041977480156651310432150449125441316478977863

Bloque 9
Intentos: 5
ID:
186033dbf595faf33c27e84b984940a90dba668f9c81806aaafed504f5c04eeeantoniomanuelrodriguezmartos6226043884
82044591396095555263373326579857536234294483213587310027014359625 25
HASH: 180ba0f6881a8b6b9afaa02f866bc55f0b50dbc17796079c0e7b7d0a995deb3b

Cadena X: 6226043884820445913960955552633733265798575362342944832135873100270143596252 5

Bloque 10
Intentos: 4
ID:
180ba0f6881a8b6b9afaa02f866bc55f0b50dbc17796079c0e7b7d0a995deb3bantoniomanuelrodriguezmartos108556887
67226423562714262409406254942010878584493969926396598787542034009311 1
HASH: 0060779b91a0a1c1f6b0f568aa1a524aa7a971fe2f32df2dfb99641054f7cbac
Cadena X: 10855688767226423562714262409406254942010878584493969926396598787542034009311 1

Bloque 11
Intentos: 7
ID:
0060779b91a0a1c1f6b0f568aa1a524aa7a971fe2f32df2dfb99641054f7cbacantoniomanuelrodriguezmartos24558932907
53217221393653521334683461949208497818658425426267006259850731888 1
HASH: 0deb776f84ff9608be209abe9281f57e934942fbc5dbe545bb9257873f2e7f98
Cadena X: 24558932907532172213936535213346834619492084978186584254262670062598507318881

Bloque 12
Intentos: 5
ID:
0deb776f84ff9608be209abe9281f57e934942fbc5dbe545bb9257873f2e7f98antoniomanuelrodriguezmartos9599842574
37160394665646900413969018361882028631269003249534760092376435298 42
HASH: 09f16f5c287d4e167d8715e761fee9edcbb11dda5fc36a063f518b0574b7e576
Cadena X: 95998425743716039466564690041396901836188202863126900324953476009237643529842

Bloque 13
Intentos: 7
ID:
09f16f5c287d4e167d8715e761fee9edcbb11dda5fc36a063f518b0574b7e576antoniomanuelrodriguezmartos9608357643
13414701516678247832934598887613018876714375262142574907229398636 51
HASH: 1cfe48f4075c9f884f2a40d5bc45a0cfb29a55045cdf255d8717139dc610c905
Cadena X: 96083576431341470151667824783293459888761301887671437526214257490722939863651

Bloque 14
Intentos: 17
ID:
1cfe48f4075c9f884f2a40d5bc45a0cfb29a55045cdf255d8717139dc610c905antoniomanuelrodriguezmartos1061503961
72527928406360979086517844680474916229733887060828357472702506992793
HASH: 07aee19c95b92ff8a4bbaae655f04ec17b2f83b3464cadc150cf1300c3d9f7da
Cadena X: 10615039617252792840636097908651784468047491622973388706082835747270250699279 3

Bloque 15
Intentos: 2
ID:
07aee19c95b92ff8a4bbaae655f04ec17b2f83b3464cadc150cf1300c3d9f7daantoniomanuelrodriguezmartos58069066280
14702144019466218226572112359808124090047434096897039155889081415 8
HASH: 06d43f3a05bf704591ab4922cf34557e9e7c04482a0499ccb83186b95860d643
Cadena X: 58069066280147021440194662182265721123598081240900474340968970391558890814158

Bloque 16
Intentos: 7
ID:
06d43f3a05bf704591ab4922cf34557e9e7c04482a0499ccb83186b95860d643antoniomanuelrodriguezmartos244069741
78981687263468129533675410406296450815107895225668379831195471777 16
HASH: 04fbc5dd776f8f044a023cb153a2e1f3336195d9ad12eeabefbc819a5ba274d9
Cadena X: 24406974178981687263468129533675410406296450815107895225668379831195471777 16

Bloque 17
Intentos: 4
ID:
04fbc5dd776f8f044a023cb153a2e1f3336195d9ad12eeabefbc819a5ba274d9antoniomanuelrodriguezmartos1847533977
32228111887747862203008990950656115199248672475637027419468897512 68

HASH: 09091430bd106fd0508bc506afac4e21fb38b3d44ee783d7195409c2431c3f45
Cadena X: 184753397732228111887747862203008990950656115199248672475637027419468897512 68

Bloque 18
Intentos: 17
ID:
09091430bd106fd0508bc506afac4e21fb38b3d44ee783d7195409c2431c3f45antoniomanuelrodriguezmartos466563030
202884345439979527376490231365811546607361079443194845594848326372 36
HASH: 06a51268f9eea9c9b7e5990bf09272f820b24ca86206a19d51bdbbbd576c2826
Cadena X: 4665630302028843454399795273764902313658115466073610794431948455948483263723 6

Bloque 19
Intentos: 7
ID:
06a51268f9eea9c9b7e5990bf09272f820b24ca86206a19d51bdbbbd576c2826antoniomanuelrodriguezmartos577198898
825930489785147964994505820177049773236736472973422159443777377969843
HASH: 1c0b31b6c2fb8e3080d4a3e88f10d8cdbf155021dbe70249683132826f692f4a
Cadena X: 57719889882593048978514796499450582017704977323673647297342215944377737969843

4.)

Utilizaremos el mismo código que el apartado anterior e iremos cambiando el valor de b.
A partir de 20 bits ya empieza a notarse cierto retardo.
Tras varias pruebas, con 22 bits ya tarda aproximadamente 3 minutos en completar la tarea.

Resultado:

b = 22
tiempo = 3 minutos

Bloque 10
Intentos: 11322191
ID:
094aa671e91d885d3776107ea6cebf580fd016f4dce59390c4a705083969de0bantoniomanuelrodriguezmartos950994243
078583283231094277282495560633256516240290972808487393968414483167 38
HASH: 0000015aa3d5038aca36ffa03f8b08740be7b43d3dd427d5c38b84eba168dbd3
Cadena X: 95099424307858328323109427728249556063325651624029097280848739396841448316738

Bloque 11
Intentos: 4501080
ID:
0000015aa3d5038aca36ffa03f8b08740be7b43d3dd427d5c38b84eba168dbd3antoniomanuelrodriguezmartos110516147
688722375873103510494875461853977003528841569466180419561281353254446
HASH: 00000387c113e7800dedfb494f20ec37486dc4f847402b28a1a4e0a7d1290489
Cadena X: 1105161476887223758731035104948754618539770035288415694661804195612813532544 46

Bloque 12
Intentos: 5879943
ID:
00000387c113e7800dedfb494f20ec37486dc4f847402b28a1a4e0a7d1290489antoniomanuelrodriguezmartos661168538
293554266580036659909704856030322773767783134001211046978541031599 2
HASH: 000001444a581512c7724f74775676b47551c847ce3ff514d0b748d037f7dc24
Cadena X: 66116853829355426658003665990970485603032277376778313400121104697854103159 92

Bloque 13
Intentos: 64421
ID:
000001444a581512c7724f74775676b47551c847ce3ff514d0b748d037f7dc24antoniomanuelrodriguezmartos114346995
089743751096119678372360430605804259562593394907090980342628324263212
HASH: 00000385d31e1d2f0de96e9d1688c9f08c62fed621037bde393551e4a3707138
Cadena X: 114346995089743751096119678372360430605804259562593394907090980342628324263212

Bloque 14
Intentos: 1546330
ID:
00000385d31e1d2f0de96e9d1688c9f08c62fed621037bde393551e4a3707138antoniomanuelrodriguezmartos441905781
756041363522343919042642239303929128566374119178132593429850343346 6
HASH: 000002d984d368a5ac375634f51e07017e5bd402bbf0159bc762a1e64147f082
Cadena X: 44190578175604136352234391904264223930392912856637411917813259342985034334 66

Bloque 15
Intentos: 3263459
ID:
000002d984d368a5ac375634f51e07017e5bd402bbf0159bc762a1e64147f082antoniomanuelrodriguezmartos952872238
551941077787295816582486659004385522327070222262804737497053479119 30
HASH: 000003687fbcba4cf69e8ffbee187314586787970d06dd24fbf5ebf1639b428e
Cadena X: 952872238551941077787295816582486659004385522327070222262804737497053479119 30

Bloque 16
Intentos: 2613193
ID:
000003687fbcba4cf69e8ffbee187314586787970d06dd24fbf5ebf1639b428eantoniomanuelrodriguezmartos5781406953
086071356317065077035974088498133326422875294866758671060995825757
HASH: 00000316809527a2b5c3047eebc9df2ecee985ff756da821620c2b894c6b97ea
Cadena X: 57814069530860713563170650777035974088498133326422875294866758671060995825757

Bloque 17
Intentos: 5931605
ID:
00000316809527a2b5c3047eebc9df2ecee985ff756da821620c2b894c6b97eaantoniomanuelrodriguezmartos1137857442
120999949734432460869922420723913172408185194009882549383150481667 04
HASH: 000003e0d6dabee401a269032647b5a92e358d355345d6e6a1b50efceba2a785
Cadena X: 1137857442120999949734432460869922420723913172408185194009882549383150481667 04

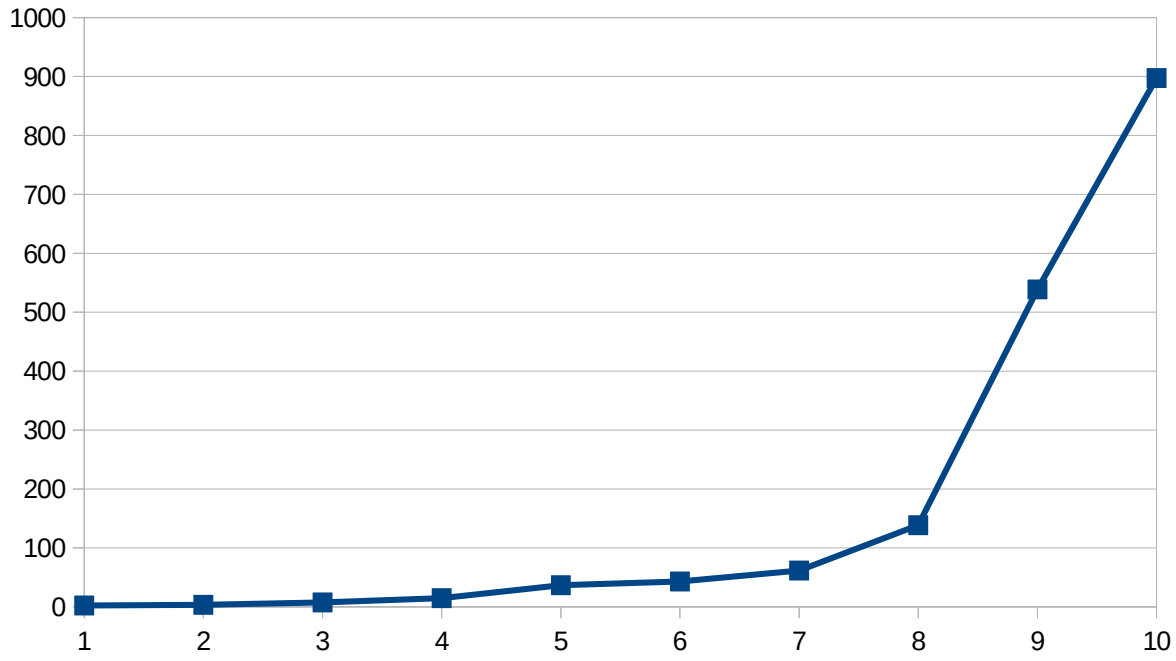
Bloque 18
Intentos: 6492711
ID:
000003e0d6dabee401a269032647b5a92e358d355345d6e6a1b50efceba2a785antoniomanuelrodriguezmartos366699871
627030333377943595231182453065492496099276438394678068675060876457 37
HASH: 000001c32d724d274ea50785654574df86313797ce5646e46388db6b4ba9662c
Cadena X: 3666998716270303333779435952311824530654924960992764383946780686750608764573 7

Bloque 19
Intentos: 1139043
ID:
000001c32d724d274ea50785654574df86313797ce5646e46388db6b4ba9662cantoniomanuelrodriguezmartos28480690
934177100828575600411094818188560547158227486250972280585714823700479
HASH: 000002e47a9a9d6127935fb03b1fcfdc3999b25e6d88b3b64240b5b749894ab5
Cadena X: 2848069093417710082857560041109481818856054715822748625097228058571482370047 9

5.)

bits	1	2	3	4	5	6	7	8	9	10
	1	5	13	18	1	72	38	223	883	928
	6	4	4	2	1	89	1	1	239	1360
	1	3	11	18	13	23	87	47	237	99
	1	10	2	8	5	72	64	20	885	2232
	4	1	15	15	94	5	23	40	1218	1408
	2	1	1	18	71	7	8	308	315	159
	3	1	3	3	2	3	44	214	334	1426
	2	1	6	31	11	56	19	294	327	991
	1	1	16	29	53	73	181	137	24	149
	1	7	4	5	115	30	150	101	924	222
media intentos	**2,2**	**3,4**	**7,5**	**14,7**	**36,6**	**43**	**61,5**	**138,5**	**538,6**	**897,4**

Tabla 1: Código apartado 1



Vemos cómo a partir de cierto valor empieza a crecer bastante comportandose como una función exponencial.

6.)

Para esta funcion modificaremos el código del apartado 1.
Ahora la generación de la nueva cadena aleatoria la haremos fuera del bucle. Ya que dentro de este, le incrementaremos de uno en uno el valor de dicha cadena si no encontramos los ceros correspondientes. Después la uniremos con el id y procederemos como hicimos anteriormente.

```
num = 256

print "Texto: "
texto = raw_input()

print "Numero de bits: "
bits = int(raw_input())

#Genaramos cadena aleatoria de num bits
cad = random.getrandbits(num)
#Concatenamos cadena con texto
id = str(cad) + texto

#Generamos nueva cadena aleatoria de num bits
nueva_cad = random.getrandbits(num)

encontrado = False
intentos = 0
while not encontrado:
    intentos += 1

    #Concatenamos el id con la nueva cadena
    nuevo_id = id + str(nueva_cad)

    #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
    hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
    #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
    cad_bin = bin(int(hash, 16))[2:].rjust(num, '0')

    #Buscamos el numero de ceros bits
    #Si no coinciden sumo 1 a la nueva cadena
    cont = 0
    for i in range(0,bits):
        if (cad_bin[i] == "0"):
            cont += 1

    if (cont == bits):
        encontrado = True
        print "Intentos:", intentos
        print "ID:", nuevo_id
        print "HASH:", hash
        print "Cadena X:", nueva_cad

    else:
        nueva_cad += 1
```

Código apartado 6

7.)

Modificaremos el código del apartado 6 definiendo la función ‘buscarCeros’ como ya hicimos en el apartado 2.

```
def buscarCeros(texto, bits, cadena):

    #Concatenamos cadena con texto
    id = str(cadena) + texto

    #Generamos nueva cadena aleatoria de num bits
    nueva_cad = random.getrandbits(num)

    encontrado = False
    intentos = 0
    while not encontrado:
        intentos += 1

        #Concatenamos el id con la nueva cadena
        nuevo_id = id + str(nueva_cad)

        #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
        hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
        #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
        cad_bin = bin(int(hash, 16))[2:].rjust(256, '0')

        #Buscamos el numero de ceros bits
        #Si no coinciden sumo 1 a la nueva cadena
        cont = 0
        for i in range(0,bits):
            if (cad_bin[i] == "0"):
                cont += 1

        if (cont == bits):
            encontrado = True
            print "Intentos:", intentos
            print "ID:", nuevo_id
            print "HASH:", hash
            print "Cadena X:", nueva_cad


        else:
            nueva_cad += 1

    return hash
```

Código apartado 7

Un resultado con los primeros diez bloques con b = 2 y los diez siguientes con b = 22 :

A partir de 20 bits ya empieza a notarse cierto retardo.

Con b = 22 tarda aproximadamente lo mismo que con el código del apartado 2. (3min)

Bloque 0
Intentos: 6
ID:
151688143592964721074135428170340438942700751870528566506366940320226774069 71antoniomanuelrodrigue
zmartos12110089947497519892823203559854943108794294272420764854430087190800893761756
HASH: 3cc0c240f27ecd68fb8c0e36fb360df257e0a1bc74f924a523f711786ff6a02f
Cadena X: 12110089947497519892823203559854943108794294272420764854430087190800893761756

Bloque 1
Intentos: 5
ID:
3cc0c240f27ecd68fb8c0e36fb360df257e0a1bc74f924a523f711786ff6a02fantoniomanuelrodriguezmartos58466726857
134396072261473932584226706483288312966649055331346308733345023989
HASH: 1cc7b7fc504c89d4ea3b8adc8f17f165e672280027674b6cdf6731693e157656
Cadena X: 58466726857134396072261473932584226706483288312966649055331346308733345023989

Bloque 2
Intentos: 4
ID:
1cc7b7fc504c89d4ea3b8adc8f17f165e672280027674b6cdf6731693e157656antoniomanuelrodriguezmartos789373489
503888610004549334857011801753125500684223496851569726487132820917 34
HASH: 067221b734df5abfc3323594c583ab2eed3438a5357b811bff14efe7b0ce5bb5
Cadena X: 78937348950388861000454933485701180175312550068422349685156972648713282091734

Bloque 3
Intentos: 1
ID:
067221b734df5abfc3323594c583ab2eed3438a5357b811bff14efe7b0ce5bb5antoniomanuelrodriguezmartos3476894975
516096866667691115354781638869013970051889416439210857320560857 85
HASH: 3709b6a028490f18a7fde55542b41b82a8af7c2dd1c6c7180e7cfaaaeb3a9b33
Cadena X: 34768949755160968666676911153547816388690139700518894164392108573205608578 5

Bloque 4
Intentos: 13
ID:
3709b6a028490f18a7fde55542b41b82a8af7c2dd1c6c7180e7cfaaaeb3a9b33antoniomanuelrodriguezmartos8383079875
704951881429986430317525452705024389216392220577567524525437570386 8
HASH: 3d73726a149e9a2344d455d3bfab25506afb3a81eeec63b42f758354a6259320
Cadena X: 83830798757049518814299864303175254527050243892163922205775675245254375703868

Bloque 5
Intentos: 1
ID:
3d73726a149e9a2344d455d3bfab25506afb3a81eeec63b42f758354a6259320antoniomanuelrodriguezmartos560007070
287626385231315879749268937007369907794150304087761139213101428019 8
HASH: 17c27f3ffac0920f205d3c3a6b5bf73c5d6ab2d4e94feec6c6127f54a4baef57
Cadena X: 56000707028762638523131587974926893700736990779415030408776113921310142801 98

Bloque 6
Intentos: 6
ID:
17c27f3ffac0920f205d3c3a6b5bf73c5d6ab2d4e94feec6c6127f54a4baef57antoniomanuelrodriguezmartos54949051349
511507567531839080735275536478873563400507191010752004355838078 06
HASH: 2f73782d4c5659d6dcea1043740a45f1c2587e39cf351acda0b411b773adca36
Cadena X: 54949051349511507567531839080735275536478873563400507191010752004355838078 06

Bloque 7
Intentos: 2

ID:
2f73782d4c5659d6dcea1043740a45f1c2587e39cf351acda0b411b773adca36antoniomanuelrodriguezmartos8364334282
6873103898977550282893585289894653388822319867060443203875572217980
HASH: 2e8de81f481f8cdfed486ac40629570a3346f6538602d455c877574b1ffecb8c
Cadena X: 836433428268731038989775502828935852898946533888223198670604432038755722179 80

Bloque 8
Intentos: 6
ID:
2e8de81f481f8cdfed486ac40629570a3346f6538602d455c877574b1ffecb8cantoniomanuelrodriguezmartos6523935220
6128028810579395846885351327279000422710718626034887152889234584924
HASH: 156561b713b109d495b026d719e6389cc862a4c942a51db962da5482e0e18c46
Cadena X: 652393522061280288105793958468853513272790004227107186260348871528892345849 24

Bloque 9
Intentos: 12
ID:
156561b713b109d495b026d719e6389cc862a4c942a51db962da5482e0e18c46antoniomanuelrodriguezmartos35837540
0610278316460897676698277702156636615935123622874897114574354108775 12
HASH: 0f5da2369ca1367d0ed6e0d6f984aaeea1a2c7d9b9265f9e59fc8229fec5308a
Cadena X: 358375400610278316460897676698277702156636615935123622874897114574354108775 12

Bloque 10
Intentos: 1891570
ID:
0f5da2369ca1367d0ed6e0d6f984aaeea1a2c7d9b9265f9e59fc8229fec5308aantoniomanuelrodriguezmartos7379498306
1604930548360867070401538024388733100008558986132934279678827234350
HASH: 0000007f7301fd2624cf36fe023df013c76c16189477cff65349c1c4b9d55ea1
Cadena X: 737949830616049305483608670704015380243887331000085589861329342796788272343 50

Bloque 11
Intentos: 96986
ID:
0000007f7301fd2624cf36fe023df013c76c16189477cff65349c1c4b9d55ea1antoniomanuelrodriguezmartos5503045150
0010765545839078147884537612663425758034866772961706685440426421810
HASH: 00000381cde20782b2771530223655f36cf4feeeba8daa6b0c1c634ed2883552
Cadena X: 550304515000107655458390781478845376126634257580348667729617066854404264218 10

Bloque 12
Intentos: 1049779
ID:
00000381cde20782b2771530223655f36cf4feeeba8daa6b0c1c634ed2883552antoniomanuelrodriguezmartos869665974
1139910767881352377688168601877793316713544493863726027857983036516 5
HASH: 0000032a671a6c4531ad83d60fe81cc46fb2c5c91b24c0566aa65319376864e3
Cadena X: 869665974113991076788135237768816860187779331671354449386372602785798303651 65

Bloque 13
Intentos: 170184
ID:
0000032a671a6c4531ad83d60fe81cc46fb2c5c91b24c0566aa65319376864e3antoniomanuelrodriguezmartos289612677
3716241551419277362533437971375633200071772547917839257736372886737 3
HASH: 00000255f6e2a185ed399edde54cb7634147e77e9b77c3c77b4b2309f3bc01c0
Cadena X: 289612677371624155141927736253343797137563320007177254791783925773637288673 73

Bloque 14
Intentos: 6398719
ID:
00000255f6e2a185ed399edde54cb7634147e77e9b77c3c77b4b2309f3bc01c0antoniomanuelrodriguezmartos493900471
0621955688615009237488963597409762464341290673795980568324913964501 1
HASH: 000001a2feafea758ef9132a6e5ca637cd9f21ede4c418b827b889958bbeec59
Cadena X: 493900471062195568861500923748896359740976246434129067379598056832491396450 11

Bloque 15

Intentos: 9134742
ID:
000001a2feafea758ef9132a6e5ca637cd9f21ede4c418b827b889958bbeec59antoniomanuelrodriguezmartos1379523011
4240218705744423396766092540632385810203687512330710137353243474750
HASH: 000001da379488fc6cef3b26c991e673b5e9952e3ddc7057db665d7a57ff4bd4
Cadena X: 13795230114240218705744423396766092540632385810203687512330710137353243474750

Bloque 16
Intentos: 2029318
ID:
000001da379488fc6cef3b26c991e673b5e9952e3ddc7057db665d7a57ff4bd4antoniomanuelrodriguezmartos179626095
2586287254377216524099608220041305270634761066015966576496840294156 8
HASH: 0000024ef7f30c4a328dc70e47904eb4641441148e7032559eeef8376b325a3f
Cadena X: 17962609525862872543772165240996082200413052706347610660159665764968402941568

Bloque 17
Intentos: 11244200
ID:
0000024ef7f30c4a328dc70e47904eb4641441148e7032559eeef8376b325a3fantoniomanuelrodriguezmartos4724721284
659801612650639799563258624419195982835641713412038756883447212296 7
HASH: 0000034902951db969860a8fbdfec4a533e4d60801985a7a0d7633b94ad91c1f
Cadena X: 472472128465980161265063979956325862441919598283564171341203875688344721229 67

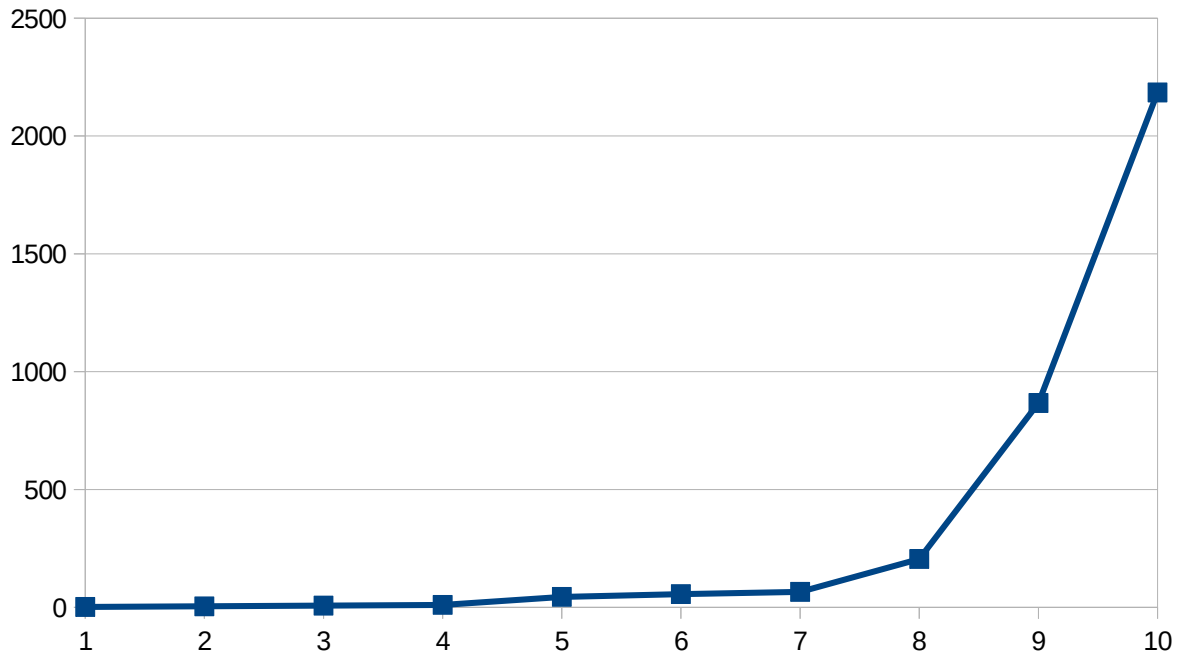
Bloque 18
Intentos: 394025
ID:
0000034902951db969860a8fbdfec4a533e4d60801985a7a0d7633b94ad91c1fantoniomanuelrodriguezmartos110047802
8783226329563936601789198280567700700308220459021212232053399840786 39
HASH: 0000005c4a30ca53204f3f2a3ffbdbf2932a45e052bcbcc0cc8fea7503ebfc43
Cadena X: 1100478028783226329563936601789198280567700700308220459021212232053399840786 39

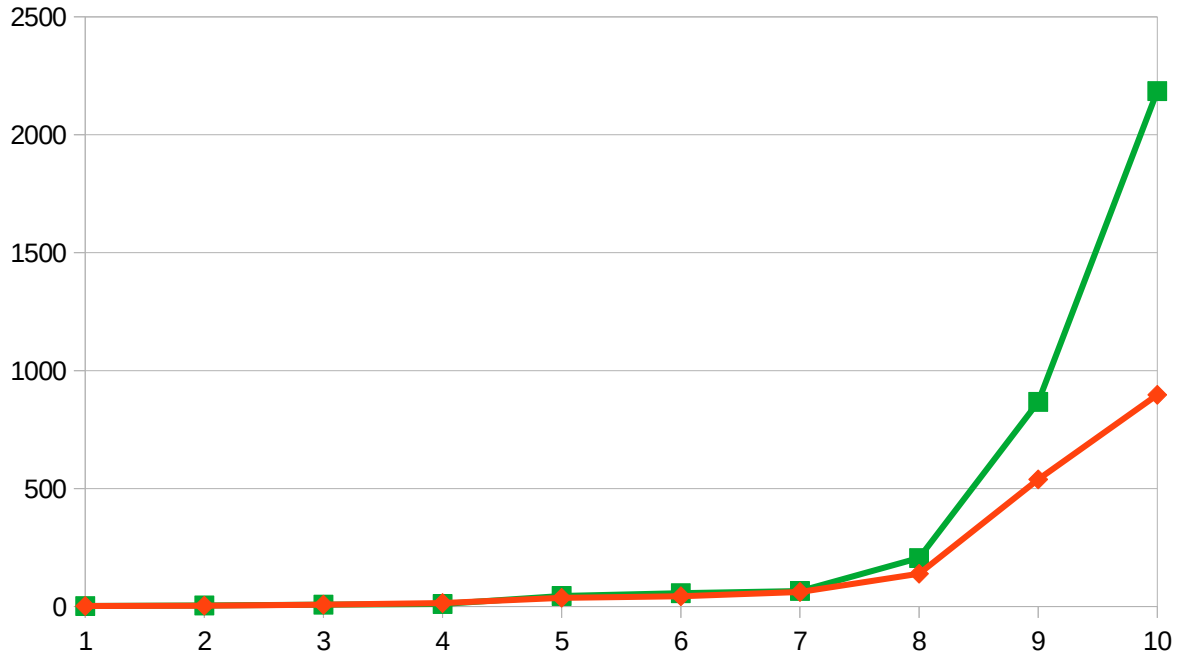
Bloque 19
Intentos: 2800741
ID:
0000005c4a30ca53204f3f2a3ffbdbf2932a45e052bcbcc0cc8fea7503ebfc43antoniomanuelrodriguezmartos85198198341
0001015111289285474755090150619634963296741053078211349899828626 29
HASH: 000002ec3a97438b70b42c2e2ba6b18715e1e4cfce32b44be3b9c52df3491584
Cadena X: 85198198341000101511128928547475509015061963496329674105307821134989982862629

8.)

bits	1	2	3	4	5	6	7	8	9	10
	1	11	2	9	18	43	39	248	2560	1706
	1	4	3	4	25	28	60	228	586	3278
	2	1	5	40	45	31	42	177	482	64
	3	12	3	5	48	86	193	50	900	3071
	2	3	4	4	5	100	65	291	2123	1686
	2	1	21	17	239	21	69	146	255	1804
	1	3	19	13	12	59	51	478	576	5163
	3	1	4	3	22	8	16	182	494	33
	1	5	2	7	21	33	56	191	32	375
	1	1	12	1	8	154	66	61	665	4670
media intentos	1,7	4,2	7,5	10,3	44,3	56,3	65,7	205,2	867,3	2185

Tabla 2: Código apartado 6





Si comparamos los dos códigos vemos como el segundo requiere de un número de intentos mayor como era de esperar.

Bibliografía

[1.] Hashlib.
https://docs.python.org/2/library/hashlib.html

[2.] SHA256 python.
https://medium.com/@dwernychukjosh/sha256-encryption-with-python-bf216db497f9

[3.] Hexadecimal python.
https://stackoverflow.com/questions/36580195/random-32-hexadecimal-digits-in-python

[4.] Hexadecimal python.
https://stackoverflow.com/questions/1425493/convert-hex-to-binary

[5.] Rellenar ceros.
https://es.stackoverflow.com/questions/60617/c%C3%B3mo-mostrar-un-n%C3%BAmero-con-ceros-a-la-izquierda

```
# -*- coding: utf-8 -*-
#! /usr/bin/env python
#EJE1

import hashlib
import random

num = 256

print "Texto: "
texto = raw_input()

print "Numero de bits: "
bits = int(raw_input())

#Genaramos cadena aleatoria de num bits
cad = random.getrandbits(num)
#Concatenamos cadena con texto
id = str(cad) + texto

encontrado = False
intentos = 0
while not encontrado:
    intentos += 1

    #Generamos nueva cadena aleatoria de num bits
    nueva_cad = random.getrandbits(num)
    #Concatenamos el id con la nueva cadena
    nuevo_id = id + str(nueva_cad)

    #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
    hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
    #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
    cad_bin = bin(int(hash, 16))[2:].rjust(num, '0')

    #Buscamos el numero de ceros bits
    cont = 0
    for i in range(0,bits):
        if (cad_bin[i] == "0"):
            cont += 1
    #Si el numero de ceros coinciden con el numero de introducido
    if (cont == bits):
        encontrado = True
        print "Intentos:", intentos
        print "ID:", nuevo_id
        print "HASH:", hash
        print "Cadena X:", nueva_cad
```

```
# -*- coding: utf-8 -*-
#! /usr/bin/env python
#EJE1.1

import hashlib
import random

num = 256

#print "Texto: "
text = "antoniomanuelrodriguezmartos"

#print "Numero de bits: "
b = 2

#Genaramos cadena aleatoria de num bits
cad = random.getrandbits(num)

def buscarCeros(texto, bits, cadena):

    #Concatenamos cadena con texto
    id = str(cadena) + texto

    encontrado = False
    intentos = 0
    while not encontrado:
        intentos += 1

        #Generamos nueva cadena aleatoria de num bits
        nueva_cad = random.getrandbits(num)
        #Concatenamos el id con la nueva cadena
        nuevo_id = id + str(nueva_cad)

        #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
        hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
        #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
        cad_bin = bin(int(hash, 16))[2:].rjust(256, '0')

        #Buscamos el numero de ceros bits
        #Si no coinciden sumo 1 a la nueva cadena
        cont = 0
        for i in range(0,bits):
            if (cad_bin[i] == "0"):
                cont += 1

        if (cont == bits):
            encontrado = True
            print "Intentos:", intentos
            print "ID:", nuevo_id
            print "HASH:", hash
            print "Cadena X:", nueva_cad

    return hash

print "------------------------------------------------------------------------"
print "Bloque 0"
hash = buscarCeros(text, b, cad)
print "------------------------------------------------------------------------"
for i in range(9):
    print "Bloque",i+1
    hash = buscarCeros(text, b, hash)
    print "------------------------------------------------------------------------"
```

```
b= 3

for i in range(10):
    print "Bloque",i+10
    hash = buscarCeros(text, b, hash)
    print "------------------------------------------------------------------------"
```

```
# -*- coding: utf-8 -*-
#! /usr/bin/env python
#EJE2

import hashlib
import random

num = 256

print "Texto: "
texto = raw_input()

print "Numero de bits: "
bits = int(raw_input())

#Genaramos cadena aleatoria de num bits
cad = random.getrandbits(num)
#Concatenamos cadena con texto
id = str(cad) + texto

#Generamos nueva cadena aleatoria de num bits
nueva_cad = random.getrandbits(num)

encontrado = False
intentos = 0
while not encontrado:
    intentos += 1

    #Concatenamos el id con la nueva cadena
    nuevo_id = id + str(nueva_cad)

    #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
    hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
    #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
    cad_bin = bin(int(hash, 16))[2:].rjust(num, '0')

    #Buscamos el numero de ceros bits
    #Si no coinciden sumo 1 a la nueva cadena
    cont = 0
    for i in range(0,bits):
        if (cad_bin[i] == "0"):
            cont += 1

    if (cont == bits):
        encontrado = True
        print "Intentos:", intentos
        print "ID:", nuevo_id
        print "HASH:", hash
        print "Cadena X:", nueva_cad

    else:
        nueva_cad += 1
```

```
# -*- coding: utf-8 -*-
#! /usr/bin/env python
#EJE2.1

import hashlib
import random

num = 256

#print "Texto: "
text = "antoniomanuelrodriguezmartos"

#print "Numero de bits: "
b = 2

#Genaramos cadena aleatoria de num bits
cad = random.getrandbits(num)

def buscarCeros(texto, bits, cadena):

    #Concatenamos cadena con texto
    id = str(cadena) + texto

    #Generamos nueva cadena aleatoria de num bits
    nueva_cad = random.getrandbits(num)

    encontrado = False
    intentos = 0
    while not encontrado:
        intentos += 1

        #Concatenamos el id con la nueva cadena
        nuevo_id = id + str(nueva_cad)

        #Calculamos el hash de nuevo id y lo guardamos como hexadecimal
        hash = hashlib.sha256(nuevo_id.encode()).hexdigest()
        #Lo pasamos a binario para encontrar los ceros, (rjust para poner todos los ceros a la izq)
        cad_bin = bin(int(hash, 16))[2:].rjust(256, '0')

        #Buscamos el numero de ceros bits
        #Si no coinciden sumo 1 a la nueva cadena
        cont = 0
        for i in range(0,bits):
            if (cad_bin[i] == "0"):
                cont += 1

        if (cont == bits):
            encontrado = True
            print "Intentos:", intentos
            print "ID:", nuevo_id
            print "HASH:", hash
            print "Cadena X:", nueva_cad
        else:
            nueva_cad += 1

    return hash


print "------------------------------------------------------------------------"
print "Bloque 0"
hash = buscarCeros(text, b, cad)
print "------------------------------------------------------------------------"
```

```
for i in range(9):
    print "Bloque",i+1
    hash = buscarCeros(text, b, hash)
    print "--------------------------------------------------------------------------"

b= 22

for i in range(10):
    print "Bloque",i+10
    hash = buscarCeros(text, b, hash)
    print "--------------------------------------------------------------------------"
```