Tecnológico de Costa Rica

Proyecto Programado 1
Criptografía y Cifrado

IC-1801
Taller de Programación
Escuela de Ingeniería en Computación

Prof. Mauricio Avilés

Estudiantes
Axel Fernández Jiménez
2016098894

María José Paz Gómez
2016106862

07/04/2016

Tabla de contenido

Manual de Usuario

Se han creado un conjunto de programas para codificar y decodificar de diferentes maneras textos y palabras. Este documento es para explicar el funcionamiento de cada uno de ellos, de forma detallada para que se eviten equivocaciones durante el proceso.

Para poder ejecutar los programas de la manera correcta, se deben realizar un conjunto de pasos:

Pasos Iniciales

1. Lo primero que se debe hacer, es descargar Python, ya que es en este lenguaje en el que se programo la información.
2. Para que Python se pueda utilizar, se debe descargar IDLE, con esta aplicación se van a poder correr los programas, de lo contrario no se podrá utilizar el documento de Python brindado.
3. Para poder usar el documento de Python que contiene los programas, se tiene que guardar en la computadora

El documento de Criptografía y Cifrados contiene varios programas, cada uno tiene su nombre en rojo y para separarlos se utilizo un conjunto de líneas (-) entre el signo de número (#). Además, cada uno de los programas tiene definido su funcionalidad y también sus entradas, salidas y restricciones para tener una mejor comprensión de cada uno de ellos.

Codificación Transposición

En este programa lo que se busca es invertir el orden de una palabra o texto asignado.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> transposicionCod(texto)
```

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar este pegado transposicionCod y además que este entre paréntesis y entre comillas. (“texto”). Si esta colocado de la manera correcta, transposicionCod y los paréntesis() van a quedar de color negro. El texto a codificar y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>transposicionCod(“texto”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> transposicionCod(“texto”)
otxet
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> transposicionCod(texto)
Debe ser un string
```

Decodificación Transposición

Aquí se quiere que lo que ya fue encriptado por este medio, pase a su manera orignal nuevamente.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> transposicionDec(texto)
```

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar este pegado transposicionDec y además que este entre paréntesis y entre comillas. (“texto”). Si esta colocado de la manera correcta, transposicionDec y los paréntesis() van a quedar de color negro. El texto a codificar y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>transposicionDec(“otxet”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> transposicionDec(“otxet”)
texto
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> transposicionDec(otext)
Debe ser un string
```

Codificación César

El cifrado César es una técnica por desplazamiento. Se sustituye cada letra del texto original y se remplaza por otra que está x posiciones más adelante. Se debe de definir x en el momento en que se define el texto que se quiere traducir.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> cesarCod(texto,desplazamiento)
```

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, y cuando se dice desplazamiento, se refiere a la cantidad de letras que debe avanzar por cada uno para llegar al nuevo valor de una letra, es muy importante que el texto que se vaya a codificar y el desplazamiento que se vaya a escoger esté pegados a cesarCod y además que estén entre paréntesis y el texto entre comillas. Hay que recordar que debe haber una coma en medio de texto y desplazamiento (“texto”,desplazamiento). Si esta colocado de la manera correcta, cesarCod , los paréntesis() y la coma(,) van a quedar de color negro. El texto a codificar, el desplazamiento y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>cesarCod(“texto”,desplazamiento)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> cesarCod(“texto”,desplazamiento(3))
whawr
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> cesarCod(texto, desplazamiento)
Debe ser un string
```

Decodificación César

Durante este proceso el texto encriptado, se quiere volver a pasar al texto original, por medio del mismo proceso.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> cesarDec(textocodificado,desplazamiento)
```

3. Donde se menciona textocodificado, se debe de colocar lo que se quiere decodificar, y cuando se dice desplazamiento, se refiere a la cantidad de letras que debe avanzar por cada uno para llegar al nuevo valor de una letra, es muy importante que el texto que se vaya a decodificar y el desplazamiento que se vaya a escoger esté pegados a cesarDec y además que estén entre paréntesis y el texto entre comillas. Hay que recordar que debe haber una coma en medio de texto y desplazamiento (“textocodificado”,desplazamiento). Si esta colocado de la manera correcta, cesarDec , los paréntesis() y la coma(,) van a quedar de color negro. El texto a decodificar, el desplazamiento y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>cesarDec(“textocodificado”,desplazamiento)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> cesarDec(“textocodificado”,desplazamiento(3)
texto
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> cesarDec(textocodificado, desplazamiento)
Debe ser un string
```

Codificación Binaria

Se quiere poner un texto normal, con letras del abecedario del uso diario, y que retorne ese texto en código binario.

Pasos

5. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

6. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> binarioCod(texto)
```

7. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar este pegado binarioCod y además que este entre paréntesis y entre comillas. (“texto”). Si esta colocado de la manera correcta, binarioCod y los paréntesis() van a quedar de color negro. El texto a codificar y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>binarioCod(“texto”)
```

8. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> binarioCod(“texto”)
10100 00100 11000 10100 01111
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> binarioCod(texto)
Debe ser un string
```

Decodificación Binaria

Al contrario de la codificación, en este caso se quiere volver a la normalidad lo que ya una vez fue encriptado en la manera binaria. Los espacios en el resultado van a ser representados de la forma “*”

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> binarioDec(texto)
```

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar este pegado binarioDec , y además que este entre paréntesis y entre comillas. (“texto”). Si esta colocado de la manera correcta, binarioDec y los paréntesis() van a quedar de color negro. El texto a codificar y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>binarioDec(“10100 00100 11000 10100 01111”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> binarioDec(“10100 00100 11000 10100 01111”)
texto
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> binarioDec(10100 00100 11000 10100 01111)
Debe ser un string
```

Codificación Leet

En el método Leet, a cada letra del abecedario se le asigna un conjunto de caracteres que se asimilen a la letra original, pero no demasiado para que no se pueda entender el código. Por ejemplo:
a = 4
b=]3
c = (

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> leetCod(texto)
```

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar este pegado leetCod y además que este entre paréntesis y entre comillas. (“texto”). Si esta colocado de la manera correcta, leetCod y los paréntesis() van a quedar de color negro. El texto a codificar y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>leetCod(“texto”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> leetCod(“texto”)
+[-)(+()
```

En este caso solo se realiza la codificación.
Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> leetCod(texto)
Debe ser un string
```

Codificación Monoalfabética

En la codificación monoalfabética se busca crear un programa en el cual a cada letra se le asigne otra letra del abecedario a partir de una palabra clave ingresada por la persona. Se pone al principio del abecedario la palabra clave elegida, se eliminan las letras repetidas dentro de la misma palabra, y las letras que ya estaban en la palabra clave, se eliminan del resto del abecedario. A partir de la ultima letra del abecedario se empiezan a poner las letras en el orden correcto, omitiendo las que ya estaban en el abecedario pasado. Cuando se codifica, si la estaba en la posición 1 en el abecedario original, y se utilizo la palabra clave hola, la a, actualmente va a ser una h, ya que es la primera letra en el abecedario nuevo, y así hasta que se terminen las letras del texto que se quiere codificar.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> monoCod(texto,palabra)
```

   siendo texto, el texto que se quiere codificar, y palabra, la palabra clave para cambiar el abecedario.

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar y la palabra clave estén pegadas a monoCod y además que estén entre paréntesis, las dos juntas, y cada uno de ellos tiene sus comillas individuales, (“texto”,”palabra”). El texto y la palabra clave siempre deben ir separadas dentro del paréntesis por una coma (,). Si esta colocado de la manera correcta, monoCod , los paréntesis(), y la coma (,) van a quedar de color negro. El texto a codificar, la palabra clave y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>monoCod(“texto”,”palabra”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> monoCod("texto","palabra")
trxtn
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> monoCod(texto,palabra)
Debe ser un string
```

Decodificación Monoalfabética

En la codificación monoalfabética se explico el mecanismo de cómo codificar de manera monoalfabética. Para la decodificación se quiere que con la palabra clave, y el texto codificado, se vuelva al texto original.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

   ```
   >>> monoDec(textocodificado,palabra)
   ```

   siendo texto, el textocodificado que se encripto en monoCod, y palabra, la palabra clave para volver el abecedario al original.

3. Donde se menciona textocodificado, se debe de colocar lo que se quiere decodificar, es muy importante que el texto que se vaya a decodificar y la palabra clave estén pegadas a monoDec y además que estén entre paréntesis, las dos juntas, y cada uno de ellos tiene sus comillas individuales (“textocodificado”,”palabra”). El texto codificado y la palabra clave siempre deben ir separadas dentro del paréntesis por una coma (,). Si esta colocado de la manera correcta, monoDec, los paréntesis(), y la coma (,) van a quedar de color negro. El texto a decodificar, la palabra clave y las comillas(“”) de un color diferente. De la siguiente manera:

   ```
   >>>monoDec(“textocodificado”,”palabra”)
   ```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

   ```
   >>> monoDec(“trxtn(textocodificado)”,”palabra”)
   texto
   ```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> monoDec(texto,palabra)
Debe ser un string
```

Codificación Vigenere

En la codificación Vigenere lo que se quiere es crear un programa en el cual también se asigne una palabra clave para cambiar el abecedario, pero de una manera diferente al monoalfabético, para poder encriptar un texto. A cada letra del abecedario, se le asigna un valor del 1 al 26, en su orden respectivo. La palabra clave va a tomar los mismos valores que en el abecedario. A cada letra del texto se le suma el valor de la letra de la palabra clave. La palabra clave se pone repetidas veces hasta que se acabe el texto.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> vigenereCod(texto,palabra)
```

siendo texto, el texto que se quiere codificar, y palabra, la palabra clave para cambiar el abecedario.

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar y la palabra clave estén pegadas a vigenereCod y además que estén entre paréntesis, las dos juntas, y cada uno de ellos tiene sus comillas individuales, (“texto”,”palabra”). El texto y la palabra clave siempre deben ir separadas dentro del paréntesis por una coma (,). Si esta colocado de la manera correcta, vigenereCod , los paréntesis(), y la coma (,) van a quedar de color negro. El texto a codificar, la palabra clave y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>vigenereCod(“texto”,”palabra”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> vigenereCod("texto","palabra")
jeitp
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> vigenereCod(texto,palabra)
Debe ser un string
```

Decodificación Vigenere

En la codificación Vigenere se explicó el mecanismo de cómo codificar de manera Vigenre. Para la decodificación se quiere que con la palabra clave, y el texto codificado, se vuelva al texto original.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

   ```
   >>> vigenereDec(textocodificado,palabra)
   ```

   siendo texto, el textocodificado que se encripto en vigenereCod, y palabra, la palabra clave para volver el abecedario al original.

3. Donde se menciona textocodificado, se debe de colocar lo que se quiere decodificar, es muy importante que el texto que se vaya a decodificar y la palabra clave estén pegadas a vigenereDec y además que estén entre paréntesis, las dos juntas, y cada uno de ellos tiene sus comillas individuales (“textocodificado”,”palabra”). El texto codificado y la palabra clave siempre deben ir separadas dentro del paréntesis por una coma (,). Si esta colocado de la manera correcta, vigenereDec, los paréntesis(), y la coma (,) van a quedar de color negro. El texto a decodificar, la palabra clave y las comillas(“”) de un color diferente. De la siguiente manera:

   ```
   >>>vigenereDec(“textocodificado”,”palabra”)
   ```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

   ```
   >>> vigenereDec(“jeitp(textocodificado)”,”palabra”)
   texto
   ```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> vigenereDec(texto,palabra)
Debe ser un string
```

Codificación Playfair

La codificación playfair se hace por medio de una sustitución con una palabra clave. Con esta se genera una matriz de letras y se usa para sustituir las letras de dos en dos, en vez de individuales. Como no alcanza, se numera 1, 2 y 3.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

```
>>> playfairCod(texto,palabra)
```

   siendo texto, el texto que se quiere codificar, y palabra, la palabra clave para cambiar el abecedario.

3. Donde se menciona texto, se debe de colocar lo que se quiere codificar, es muy importante que el texto que se vaya a codificar y la palabra clave estén pegadas a playfairCod y además que estén entre paréntesis, las dos juntas, y cada uno de ellos tiene sus comillas individuales, (“texto”,”palabra”). El texto y la palabra clave siempre deben ir separadas dentro del paréntesis por una coma (,). Si esta colocado de la manera correcta, playfairCod , los paréntesis(), y la coma (,) van a quedar de color negro. El texto a codificar, la palabra clave y las comillas(“”) de un color diferente. De la siguiente manera:

```
>>>playfairCod(“texto”,”palabra”)
```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

```
>>> playfairCod(“texto”,”palabra”)
extet2
```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> playfairCod(texto,palabra)
Debe ser un string
```

Decodificación Playfair

La decodificación de playfair hace que la palabra que ya fue codificada pase a su estado normal de nuevo.

Pasos

1. Se debe poner a correr el programa, esto se hace cuando ya se está en el documento de Criptografía y Cifrados y se pulsa la tecla F5, esto hace que aparezca el shell de Python.

2. Para este programa, se debe poner en el primer renglón del shell de Python

   ```
   >>> playfairDec(textocodificado,palabra)
   ```

   siendo textocodificado, el texto que se quiere decodificar, y palabra, la palabra clave para cambiar el abecedario.

3. Donde se menciona texto, se debe de colocar lo que se quiere decodificar, es muy importante que el texto que se vaya a decodificar y la palabra clave estén pegadas a playfairDec y además que estén entre paréntesis, las dos juntas, y cada uno de ellos tiene sus comillas individuales, (“textocodificado”,”palabra”). El texto y la palabra clave siempre deben ir separadas dentro del paréntesis por una coma (,). Si esta colocado de la manera correcta, playfairDec , los paréntesis(), y la coma (,) van a quedar de color negro. El texto a codificar, la palabra clave y las comillas(“”) de un color diferente. De la siguiente manera:

   ```
   >>>playfairDec(“textocodificado”,”palabra”)
   ```

4. A continuación se presiona RETURN o ENTER y esto va a hacer que le devuelva el mensaje que se asignó, de la manera encriptada. De la siguiente manera:

   ```
   >>> playfairDec(“extet2(textocodificado)”,”palabra”)
   texto
   ```

Si hubo algún error durante el procedimiento anterior, el programa va a decir que debe ser un string, por ejemplo:

```
>>> playfairDec(texto,palabra)
Debe ser un string
```