## Requerimientos

Windows, Mac OSX o cualquier otra plataforma con Python 3.5.1 instalado.

<https://www.python.org/downloads/>

## Objetivo

Desarrollar un programa que, dado un número en base 10 con punto decimal, obtenga su representación en punto flotante de simple precisión y dado un numero en punto flotante de simple precisión obtenga su valor en base diez con punto decimal.

## Pseudocódigos

**dec2bin (n, lista)**

lista 🡨 n mod 2

n 🡨 n/2

SI n!=1 y n!=0

dec2bin (n, lista)

REGRESA lista.invertida()

**bin2dec (lista)**

DESDE i🡨0 HASTA tamañoLista

num = num + lista[i]\*2i

REGRESA num

**frac2bin (decimal)**

MIENTRAS tamañoLista2 < 23 y decimal != 0

entero = decimal \* 2

lista2 🡨 lista2 (concatenado )entero

decimal = decimal - entero

REGRESA lista2

**decimalpart(binary)**

DESDE I 🡨 1 HASTA i < 10

Lista[i] 🡨 binary[i]

exp = bin2dec(Lista) – 127

fraccion=1

DESDE i 🡨 9 y j 🡨 1HASTA i 🡨 32 y j 🡨 23

fraccion= fracción + binary(i)\*

REGRESA (-1^binary(0))\*fracción\*(2^exp)

**flotante(numero)**

SI numero < 0

signo=1

SI NO

signo=0

ent = dec2bin(numero.parteEntera(),lista)

dec = frac2bin(numero.positivo() – numero.parteEntera())

expo = dec2bin(127+entero.tam,lista2)

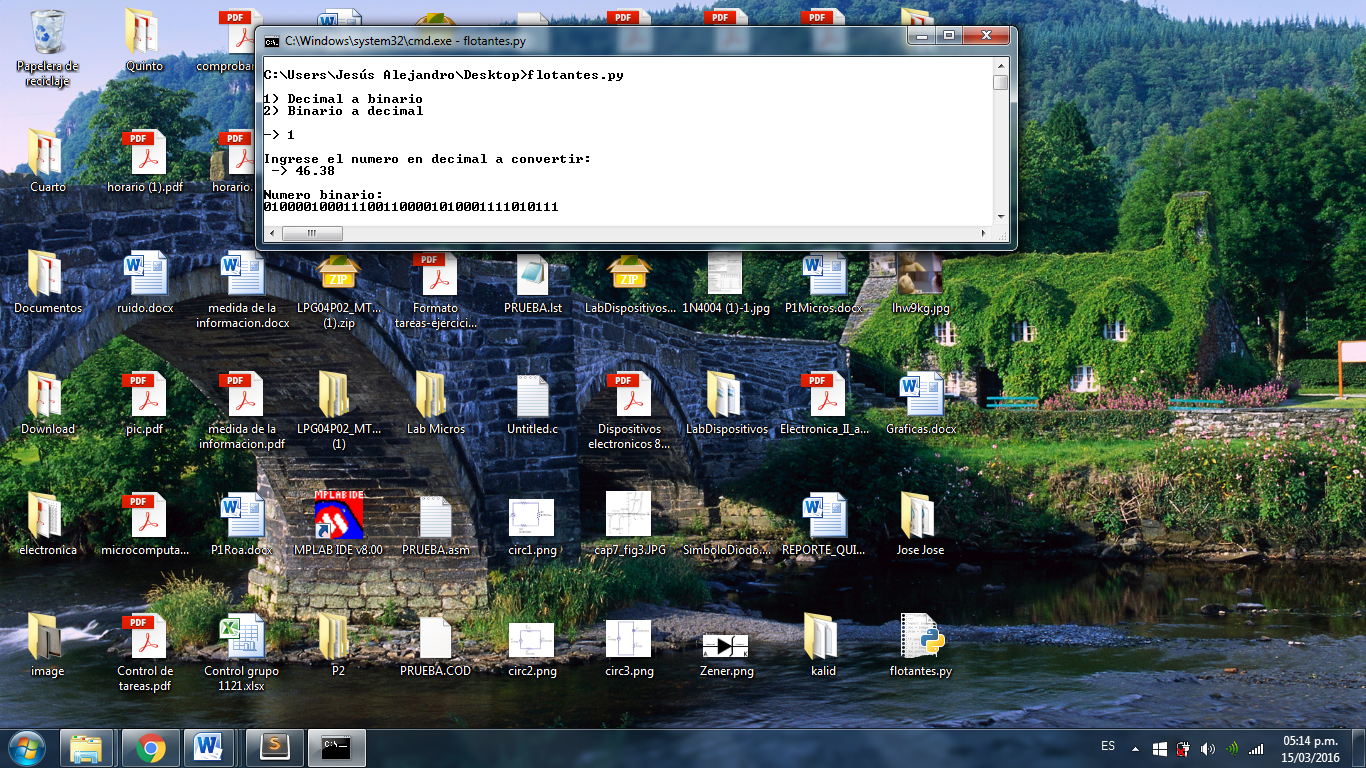
REGRESA lista.concatenar(signo,exp,ent[1:],dec)

## Modo de uso.

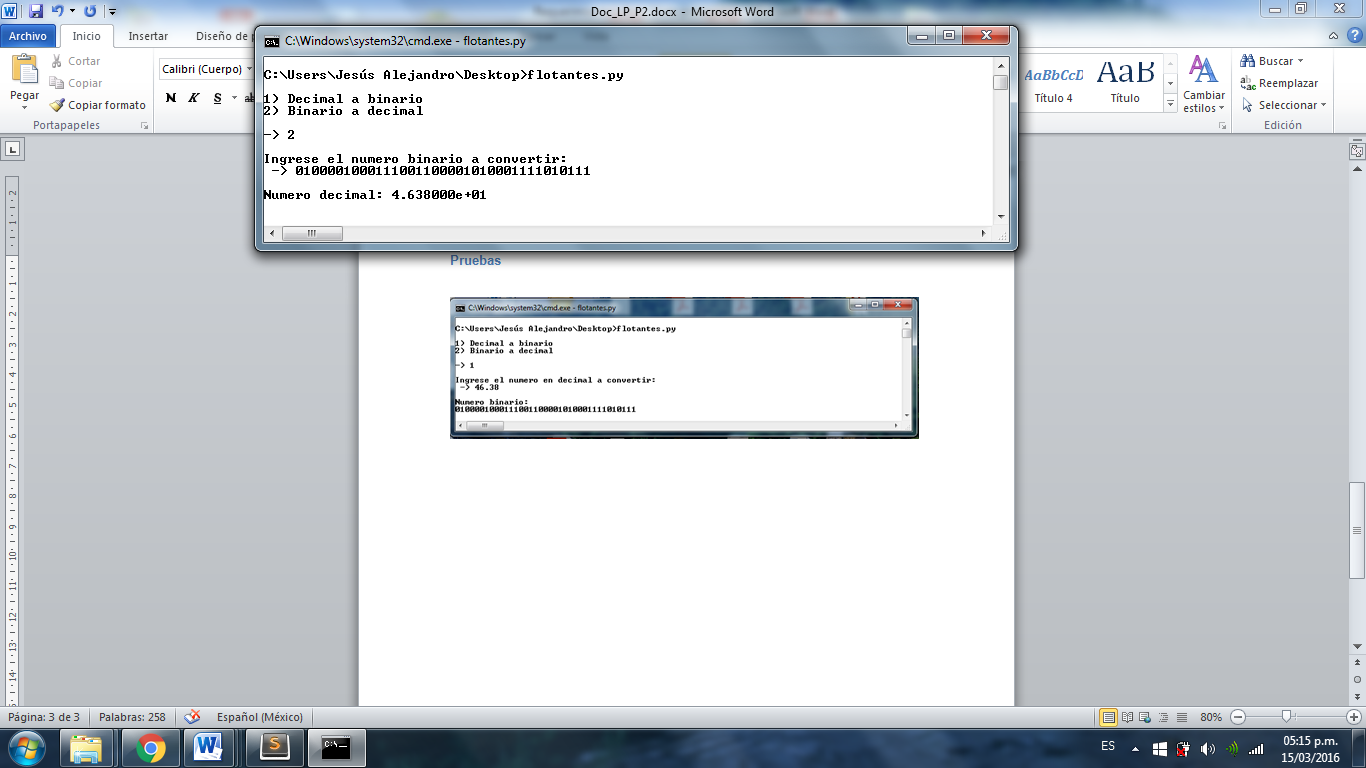
1. Abrir el programa y elegir si se desea convertir de un numero base 10 con punto decimal a binario o un numero binario a su representación en base 10
2. Posteriormente se debe ingresar el número según la opción seleccionada, si se trata del binario al decimal se deben ingresar los 32 bits correspondientes a la notación de simple precisión
3. Una vez hecho lo anterior, se puede ver en pantalla el resultado equivalente en la representación seleccionada

## Pruebas

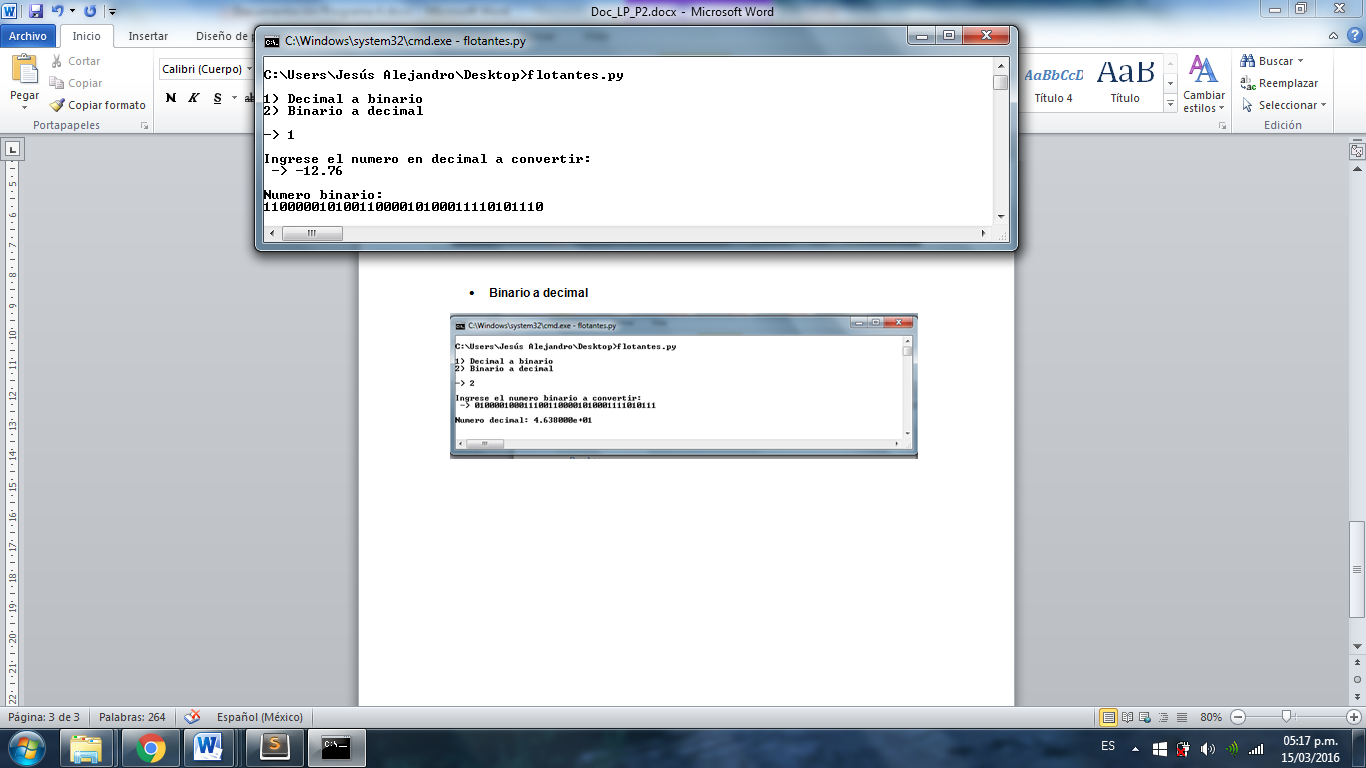
* **Decimal a binario**



* **Binario a decimal**



* **Decimal negativo a binario**



* **Binario de un numero negativo a decimal**

