**CONTROL DE ERRORES**

1. Define las etiquetas:
   1. Try.
   2. Except.
   3. Finally.
   4. assert
2. A continuación se muestran dos ejemplos de implementación de un programa que calcula la serie de Fibonacci para números menores a 20.

En primer lugar se muestra una implementación sin uso de excepciones, con las herramientas vistas antes de esta unidad.

def calcularFibonacciSinExcepciones(n):

if (n>=20) or (n<=0):

print ''' Ha ingresado un valor incorrecto.

El valor debe ser número entero mayor a cero y menor a 20'''

return

salida=[]

a,b = 0,1

for x in range(n):

salida.append(b)

a, b = b, a+b

return salida

def mainSinExcepciones():

input = raw\_input('Ingrese n para calcular Fibonacci(n):')

n = int(input)

print calcularFibonacciSinExcepciones(n)

1. A continuación un código que utiliza excepciones para manejar la entrada de mejor manera.

def calcularFibonacciConExcepciones(n):

try:

assert(n>0)

assert(n<20)

except AssertionError:

raise ValueError

a=0

b=1

salida = []

for x in range(n):

salida.append(b)

a, b = b, a+b

return salida

def mainConExcepciones():

try:

input = raw\_input('Ingrese n para calcular Fibonacci(n):')

n = int(input)

print calcularFibonacci2(n)

except ValueError:

print '''Ha ingresado un valor incorrecto.

El valor debe ser un número entero mayor a cero y menor a 20'''