Proyecto 2, Modelos de Gestión Financiera

Sebastian Puerto

25 de octubre de 2019

Punto 1

In [2]:
```
from simulador_S import grafico_valor_activo # Funcion simuladora creada en archivo separado

import numpy as np
import matplotlib.pyplot as plt
import time
%matplotlib inline
```

Función para simulación

El precio de una acción está siendo modelada por la ecuación diferencial estocástica $dS = \mu S dt + \sigma S dW$, con W un proceso de Wiener. El precio de la acción la estamos aproximando tomando dt "lo suficientemente pequeños" y hallando el dS correspondiente.

    Parámetros

S0: precio en el tiempo 0 del activo

mu: rentabilidad promedio del activo

sig: volatilidad del activo

dt: cambio "infinitesimal" de tiempo. Afecta la precision de las aproximaciones.

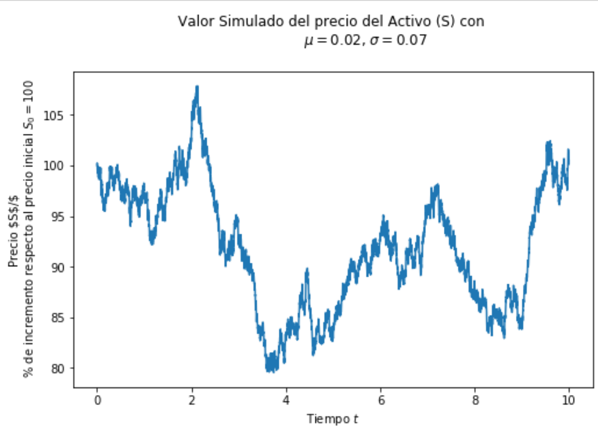
Dt: periodo de gran escala de tiempo, correspondiente a 1 unidad de tiempo

N: numero de periodos o tiempos a graficar graficar: booleano indicando si se quiere o no graficar los datos hallados

pts: número de parejas (t, Ss) subconjunto del total calculado que se desean retornar y graficar (i.e. resolución e los datos retornados)

txtad: texto adicional a imprimir en el gráfico

In [11]:
```
grafico_valor_activo(S0 = 100, mu = 0.02, sig = 0.07, dt = 0.001, Dt = 1, N = 10, graficar = True, pts = 0, txtad = "")
```



Out[11]:
```
(array([0.000e+00, 1.000e-03, 2.000e-03, ..., 9.998e+00, 9.999e+00,
        1.000e+01]),
 array([100.        , 100.24523978, 100.21872879, ..., 100.16475906,
        100.11972553, 100.46219539]))
```

Estimacion de precisión necesaria

Para saber cuánto debe ser el intervalo de timpo "infinitesimal" dt, hallamos el error porcentual al calcular la función $e^{0.9*48}$, donde 0.9 representa el promedio y 48 el tiempo (por ejemplo meses). De aquí concluimos que con un $dt = 0.001$ se obtiene un buen equilibrio entre tiempo y precisión de la solución de la ecuación diferencial.

In [2]:
```
def est_precis(ddt, mmu = 0.9, Ene = 48):
    start = time.time()
    ts, Ss = grafico_valor_activo(S0 = 1, mu = mmu, sig = 0, dt = ddt, Dt = 3, N = Ene, graficar = False)
    t = ts[-1]
    exCalc = Ss[-1]
    exReal = np.exp(mmu*t)
    error = np.abs(exCalc/exReal - 1)
    print("Para dt =", ddt, ", e**(", mmu, t, ") calculado:\t", exCalc)
    print("Para dt =", ddt, ", e**(", mmu, t, ") real:\t", exReal)
    print("Asi que, para dt =", ddt, " hubo un error de:", error*100, "%")
    print("Tiempo: ", time.time() - start, "\n")
est_precis(ddt = 0.0001)
est_precis(ddt = 0.0005)
est_precis(ddt = 0.001)
est_precis(ddt = 0.01)
```

```
Para dt = 0.0001 , e**( 0.9 144.0 ) calculado:   1.9143978647135023e+56
Para dt = 0.0001 , e**( 0.9 144.0 ) real:        1.9255945791484567e+56
Asi que, para dt = 0.0001  hubo un error de: 0.5814679037944637 %
Tiempo:  10.732323408126831

Para dt = 0.0005 , e**( 0.9 144.0 ) calculado:   1.8702713688723112e+56
Para dt = 0.0005 , e**( 0.9 144.0 ) real:        1.9255945791484567e+56
Asi que, para dt = 0.0005  hubo un error de: 2.8730455971999413 %
Tiempo:  2.2553927898406982

Para dt = 0.001 , e**( 0.9 144.0 ) calculado:    1.816569369920516e+56
Para dt = 0.001 , e**( 0.9 144.0 ) real:         1.9255945791484567e+56
Asi que, para dt = 0.001  hubo un error de: 5.661898429115553 %
Tiempo:  1.14546537399292

Para dt = 0.01 , e**( 0.9 144.0 ) calculado:     1.0784345960075816e+56
Para dt = 0.01 , e**( 0.9 144.0 ) real:  1.9255945791484567e+56
Asi que, para dt = 0.01  hubo un error de: 43.99472206218555 %
Tiempo:  0.10913729667663574
```

Ejemplos de Precio de Acción Simulados

Ejemplo con varias simulaciones: $\mu = 0.031$

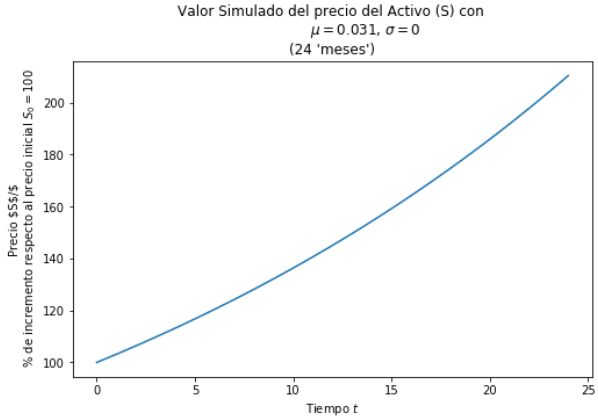
Simulamos el precio de un activo con rendimiento promedio de 0.031 para distintos sigmas para analizar su comportamiento. Cada simulación la realizamos 2 veces puesto que incluso con exactamente los mismos parámetros se pueden obtener evoluciones del precio distintas. Además, usamos dos periodos de tiempo distintos para las simulaciones: 24 y 72 (que interpretamos como meses debido a la magnitud de μ) para ver las posibles diferencias.

Eligiendo el precio como $S_0 = 100$ **podemos interpretar el eje y de las gráficas como el valor porcentual del activo respecto al precio inicial**. En general cambiar S_0 solo cambiar la escala del eje y, nada más se ve afectado, pues la ecuación habla del cambio proporcional en S, $\frac{dS}{S}$.

$\mu = 0.031$

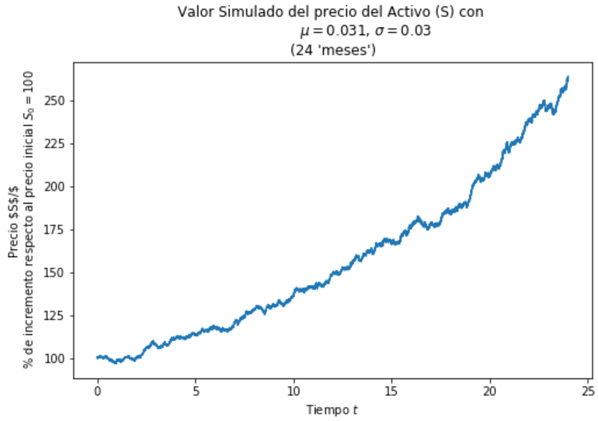
$\sigma = 0$

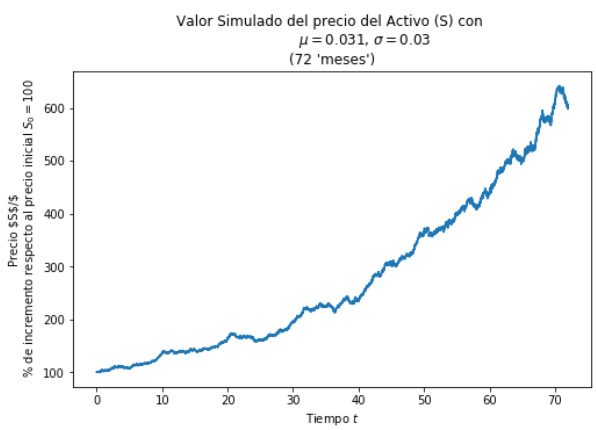
In [3]:
```
# Mes
_= grafico_valor_activo(mu = 0.031, sig = 0, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0, Dt = 1, N = 24*3, txtad = "(72 'meses')")
```



$\sigma = 0.03$

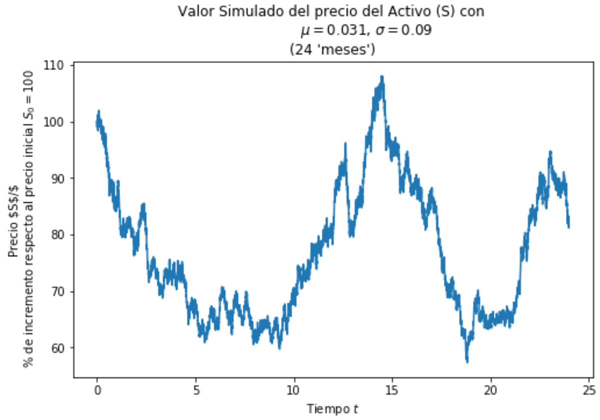
In [4]:
```
# Mes
_= grafico_valor_activo(mu = 0.031, sig = 0.03, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.03, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.03, Dt = 1, N = 24*3, txtad = "(72 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.03, Dt = 1, N = 24*3, txtad = "(72 'meses')")
```





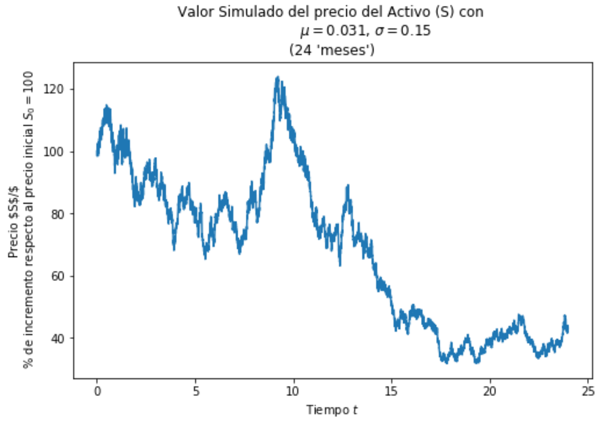
$\sigma = 0.09$

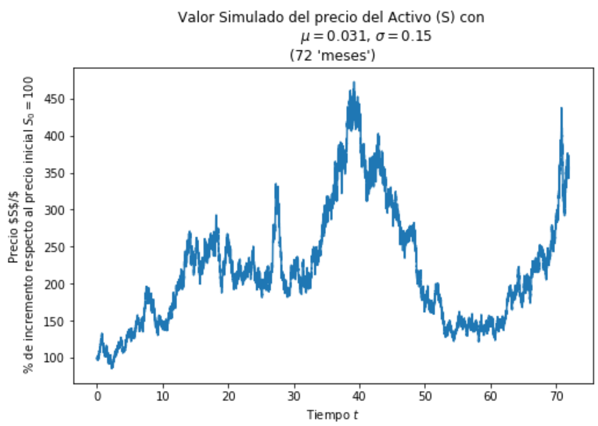
In [5]:
```python
# Mes
_= grafico_valor_activo(mu = 0.031, sig = 0.09, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.09, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.09, Dt = 1, N = 24*3, txtad = "(72 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.09, Dt = 1, N = 24*3, txtad = "(72 'meses')")
```

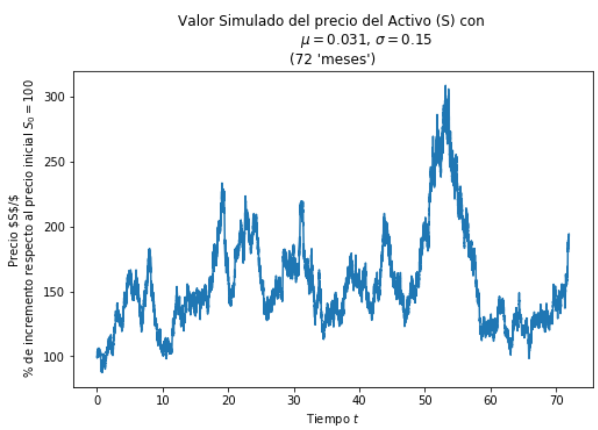


$\sigma = 0.15$

In [6]:
```python
# Mes
_= grafico_valor_activo(mu = 0.031, sig = 0.15, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.15, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.15, Dt = 1, N = 24*3, txtad = "(72 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.15, Dt = 1, N = 24*3, txtad = "(72 'meses')")
```

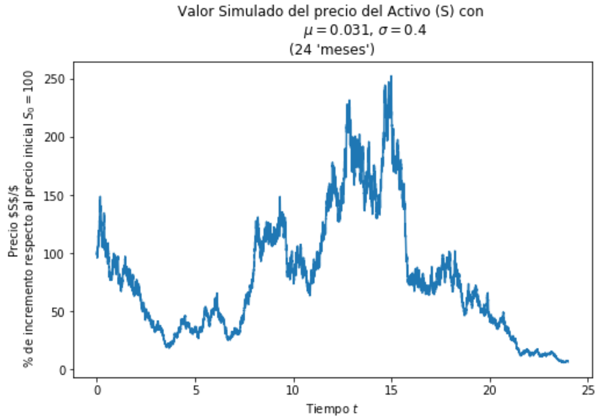






$\sigma = 0.4$

In [7]:
```python
# Mes
_= grafico_valor_activo(mu = 0.031, sig = 0.4, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.4, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.4, Dt = 1, N = 24*3, txtad = "(72 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 0.4, Dt = 1, N = 24*3, txtad = "(72 'meses')")
```



$\sigma = 1$

```
In [8]: # Mes
_= grafico_valor_activo(mu = 0.031, sig = 1, Dt = 1, N = 24, txtad = "(24 'meses')")
_= grafico_valor_activo(mu = 0.031, sig = 1, Dt = 1, N = 24*3, txtad = "(72 'meses')")
```

Análisis para $\mu = 0.031$

Observamos que

- Si σ es de una magnitud similar, aunque algo mayor, a μ ($\sigma = 0.09$) se obtienen los resultados más interesantes. En este caso ocurren que el precio puede tanto subir significativamente, como mantenerse cercano al valor original en promedio pero experimentando cambios muy erráticos. Para esta pareja de parámetros es muy complicado saber el posible comportamiento del precio, ningún parámetro domina al otro.
- Si σ es de una magnitud menor a μ, el comportamiento exponencial dado por $S = S_0 e^{\mu t}$ domina el comportamiento del precio de S, pero se observan imperfecciones en esta curva, cuyo tamaño dependen del tamaño exacto de σ.
- Si el riesgo es muy algo comparado con μ, aunque inicialmente se pueden llegar a observar incrementos más que significativos en el precio de hasta el 600% ($\sigma = 0.4$), al igual que disminuciones muy radicales, el precio en algún momento llega a ser 0, y a partir de este punto no puede dejar de tomar este valor.
- Observamos que la vista general de las gráficas de 24 y 72 periodos de tiempo es muy similar, hay un comportamiento de fractal en estas gráficas.

Otros μ's

Hacemos unas cuantas de simulaciones con otros μs para llevar a cabo un análisis teniendo en cuenta esta variación de parámetros.

Lo que pudimos concluir es que las conclusiones anteriores para $\mu = 0.031$ se siguen manteniendo. Obseravmos también que el la escala de precios varía exponencialmente al cambiar el orden de magnitud e μ, como es de esperarse por el término $e^{\mu t}$. También observamos que el comportamiento es algo más radical entre más se incrementa μ y de igual forma σ (por ejemplo, $\mu = 1$, $\sigma = 0.8$), llegando a un porcentaje de incremente de 10^{16} en un periodo de 48 periodos, de nuevo, debido al término exponencial, el cual hace invisible la importancia de lo ocurrido anteriormente.

Esto nos muestra que cada μ tiene sentido en cierta escala de tiempo.

```
In [3]: def plot_texto(texto, taman):
    plt.figure(figsize = (3,1))
    plt.axis('off')
    plt.text(0, 0, texto, fontsize=taman)
```

In [4]:

```
mus = [0.02, 0.2, 1]
sigs = [0, 0.005, 0.05, 0.4, 0.8, 1.1, 2]

for mmu in mus:
    plot_texto("$\mu$ = " + str(mmu), 30)

    for ssig in sigs:
        plot_texto("$\sigma$ = " + str(ssig), 18)
        _= grafico_valor_activo(mu = mmu, sig = ssig, Dt = 1, N = 24*2, txtad = "(48 periodos)")
```

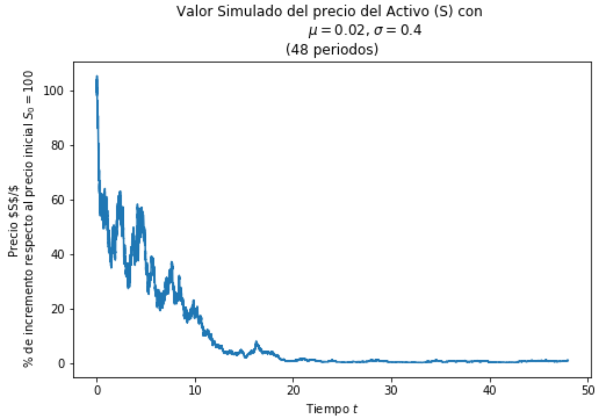
μ = 0.02

σ = 0

σ = 0.005

σ = 0.05

σ = 0.4



$\sigma = 0.8$

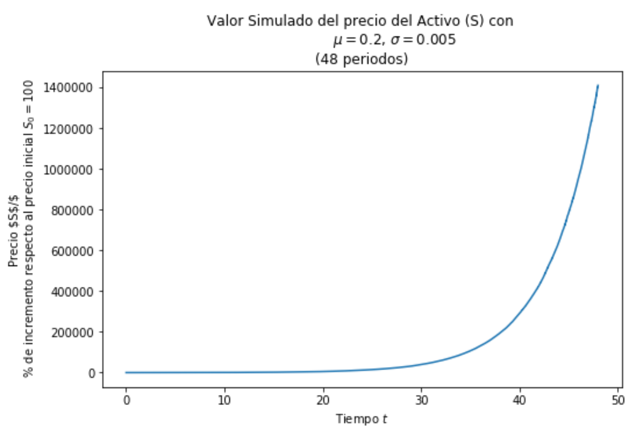
$\sigma = 1.1$

$\sigma = 2$

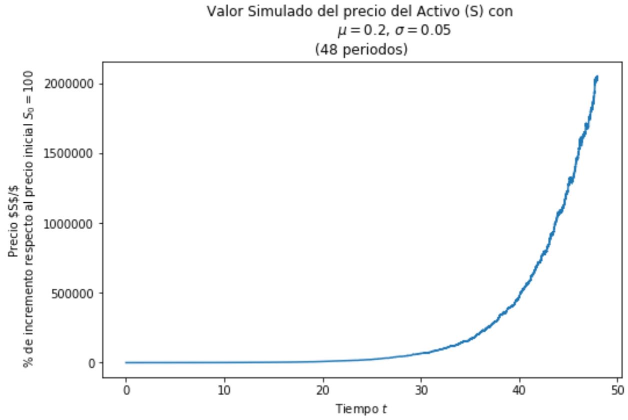
$\mu = 0.2$

$\sigma = 0$

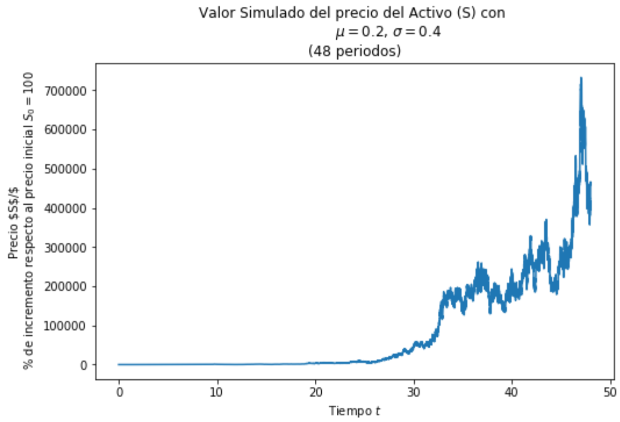
$\sigma = 0.005$



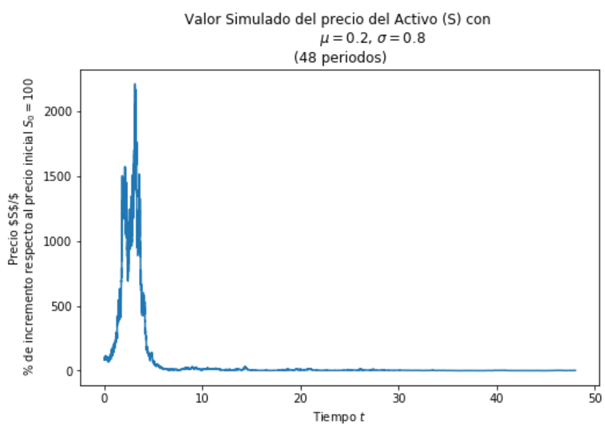
$\sigma = 0.05$



$\sigma = 0.4$



$\sigma = 0.8$

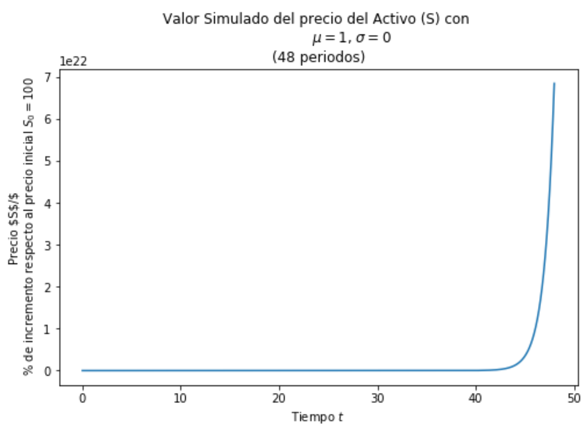


$\sigma = 1.1$

$\sigma = 2$

$\mu = 1$

$\sigma = 0$



$\sigma = 0.005$

$\sigma = 0.05$

$\sigma = 0.4$

$\sigma = 0.8$

$\sigma = 1.1$

$\sigma = 2$