



Nombre de la práctica	Manual de Practicas Regresión Lineal en Anaconda			No.	1
Asignatura:	Simulación 3501	Carrera:	Ingeniería en Sistemas Computacionales	Duración de la práctica (Hrs)	8

I. Competencia(s) específica(s): **Alumno: Raúl Ciriaco Castillo 3501**

**Github:**

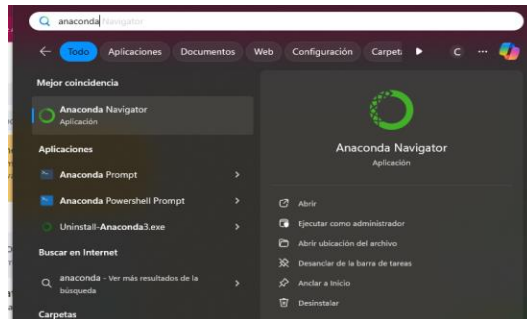
**Comentado [T1]:** <https://github.com/RaulCiriaco>

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

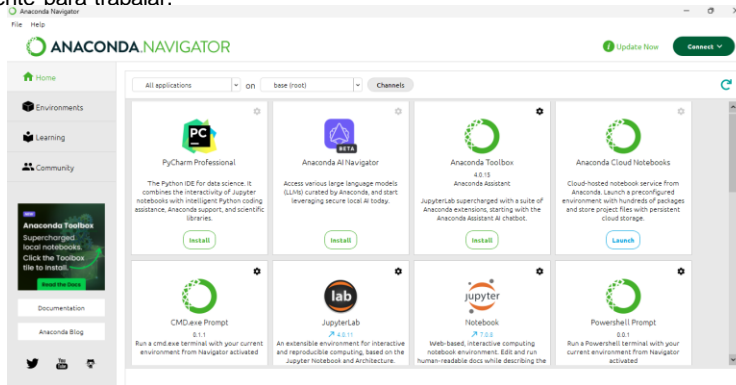
III. Material empleado:

**Desarrollo de la práctica:**

Como primeros pasos debemos instalar Anaconda Studio en nuestro Equipo de Computo que vamos a utilizar



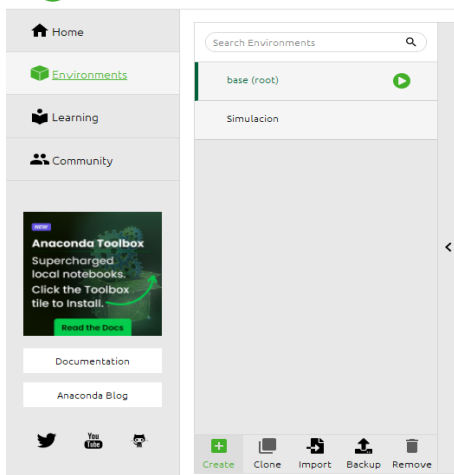
Una vez que instalamos las paqueterías y todo lo requerido abrimos Anaconda y procedemos a crear nuestro ambiente para trabajar.





File Help

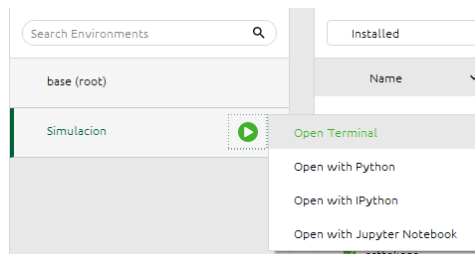
ANACONDA.NAVIGATOR



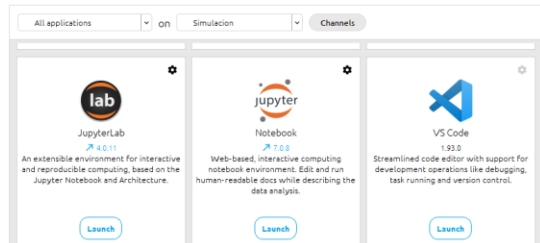
Debemos dar click en el apartado "Environments" y abajo pulsamos el "Create"

Después pulsamos en Open Terminal y vamos a proceder a instalar las paqueterías:

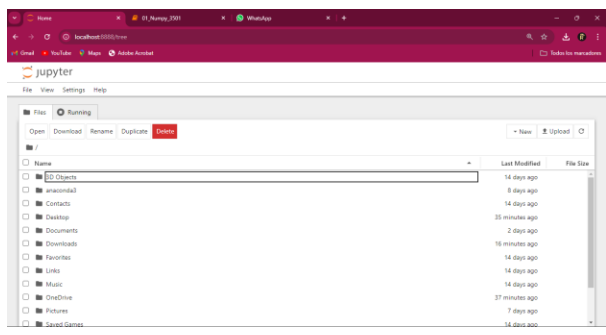
- Numpy
- Pandas
- Matplotlib
- SkLearn



Vamos a descargar el Jupyter Notebook y pulsamos en Launch.



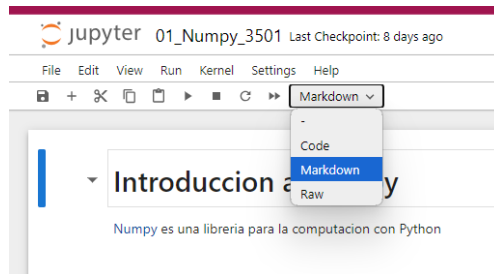
Nos abrirá esta pantalla donde podremos navegar libremente, crear archivos, modificar y eliminarlos.





### Crearemos un nuevo archivo para comenzar a trabajar el Tema: Regresión Lineal: Costo de un incidente de seguridad

En este apartado principal tendremos las herramientas que podemos utilizar en nuestro archivo, tenemos las opciones que se muestran en la imagen



- **Code:** Este apartado nos sirve para poder realizar instrucciones que queramos ejecutarlas después, que cumplan con alguna condición o no, cualquier actividad que tengamos que codificar se verá reflejada en este espacio.
- **Markdown:** Este nos permite la escritura de cualquier información, ya sean formulas o códigos para ejemplificar, no funciona si queremos mostrar o imprimir algún dato o valor.

Comenzaremos con una línea que nos describirá el problema a solucionar en esta problemática

### Regresión Lineal: Costo de un incidente de seguridad

En este ejercicio se muestran los fundamentos básicos de la regresión lineal aplicada a un caso de uso sencillo relacionado con la ciberseguridad.)

#### Enunciado del ejercicio

El ejercicio consiste en predecir el costo de un incidente de seguridad en base al número de equipos que se han visto afectados. El conjunto de datos es generado de manera aleatoria



## 1.- Generación del DataSet

```
] : import numpy as np
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)

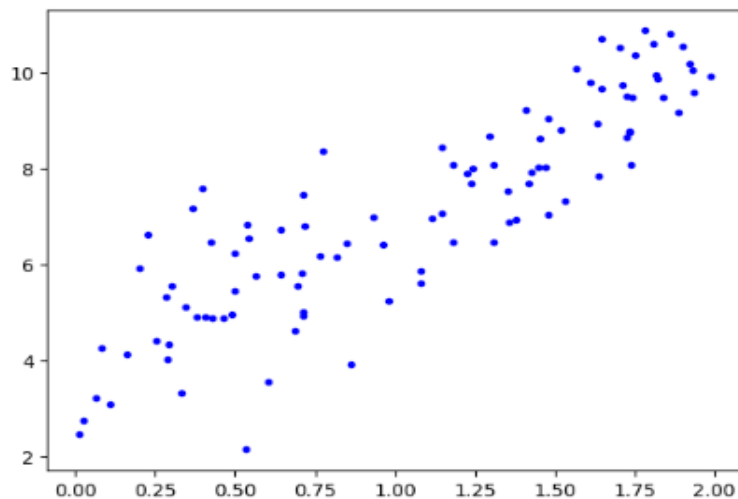
print("La longitud del DataSet es: ", len(X))

La longitud del DataSet es: 100
```

## 2.- Visualización del DataSet

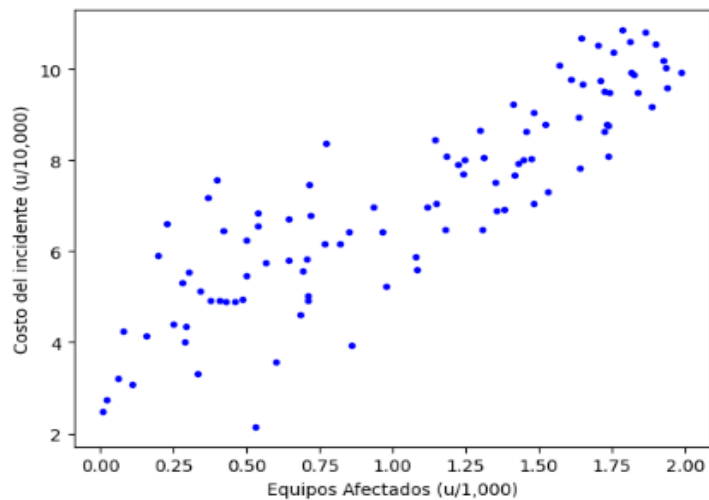
```
2]: import matplotlib.pyplot as plt
%matplotlib inline
```

```
3]: plt.plot(X, y, "b.")
plt.show()
```





```
[4]: plt.plot(X, y, "b.")
plt.xlabel("Equipos Afectados (u/1,000)")
plt.ylabel("Costo del incidente (u/10,000)")
plt.show()
```



## Modificación del DataSet

```
[5]: import pandas as pd

[6]: data = {'No_Equipos_Afectados': X.flatten(), 'Costo': y.flatten()}
df = pd.DataFrame(data)
df.head(10)
```

```
[6]:
```

	No_Equipos_Afectados	Costo
0	0.772927	8.365254
1	1.809500	10.595150
2	0.693590	5.570196
3	0.711309	5.012541
4	1.737080	8.078758
5	0.643902	5.793228
6	0.978156	5.240593
7	1.081523	5.605819
8	1.637875	7.835306
9	1.931732	10.043583

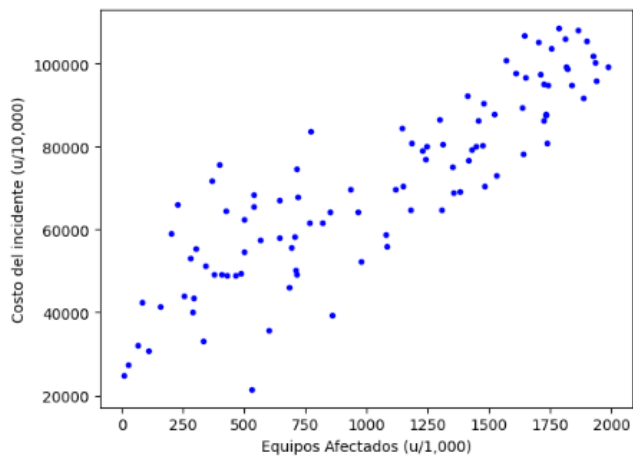


```
[7]: # Escalado del Número de equipos afectados
df['No_Equipos_Afectados'] = df['No_Equipos_Afectados'] * 1000
df['No_Equipos_Afectados'] = df['No_Equipos_Afectados'].astype('int')
# Escalado del Costo
df['Costo'] = df['Costo'] * 10000
df['Costo'] = df['Costo'].astype('int')
df.head(10)
```

```
[7]:
```

	No_Equipos_Afectados	Costo
0	772	83652
1	1809	105951
2	693	55701
3	711	50125
4	1737	80787
5	643	57932
6	978	52405
7	1081	56058
8	1637	78353
9	1931	100435

```
[8]: plt.plot(df['No_Equipos_Afectados'], df['Costo'], "b.")
plt.xlabel("Equipos Afectados (u/1,000)")
plt.ylabel("Costo del incidente (u/10,000)")
plt.show()
```





## 4.- Construcción del Modelo

```
[9]: from sklearn.linear_model import LinearRegression

[10]: # Construcción del modelo y ajuste de la función de hipótesis
lin_reg = LinearRegression()
lin_reg.fit(df['No_Equipos_Afectados'].values.reshape(-1, 1), df['Costo'].values)

[10]: + LinearRegression ⓘ ⓘ
LinearRegression()

[11]: # Parámetro Theta 0
lin_reg.intercept_

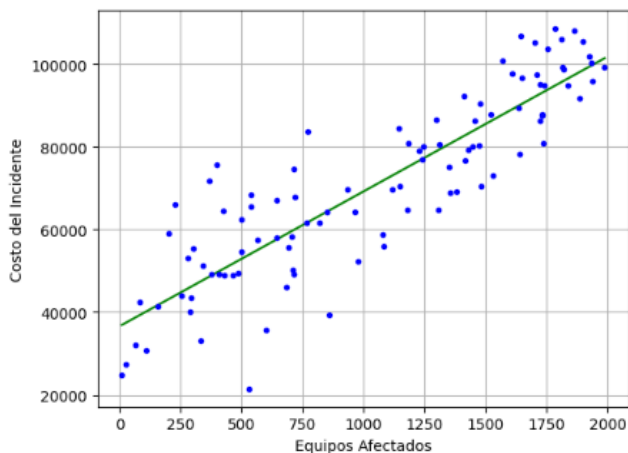
[11]: np.float64(36600.04799018018)

[12]: # Parámetro Theta 1
lin_reg.coef_

[12]: array([32.64290451])

[13]: # Predicción para el valor máximo, mínimo para el conjunto de datos de entrenamiento
X_min_max = np.array([[df['No_Equipos_Afectados'].min()], [df['No_Equipos_Afectados'].max()]])
y_train_pred = lin_reg.predict(X_min_max)

[14]: # Representación Gráfica de La Función de Hipótesis generada
plt.plot(X_min_max, y_train_pred, "g-")
plt.plot(df['No_Equipos_Afectados'], df['Costo'], "b.")
plt.xlabel("Equipos Afectados")
plt.ylabel("Costo del Incidente")
plt.grid(True)
plt.show()
```



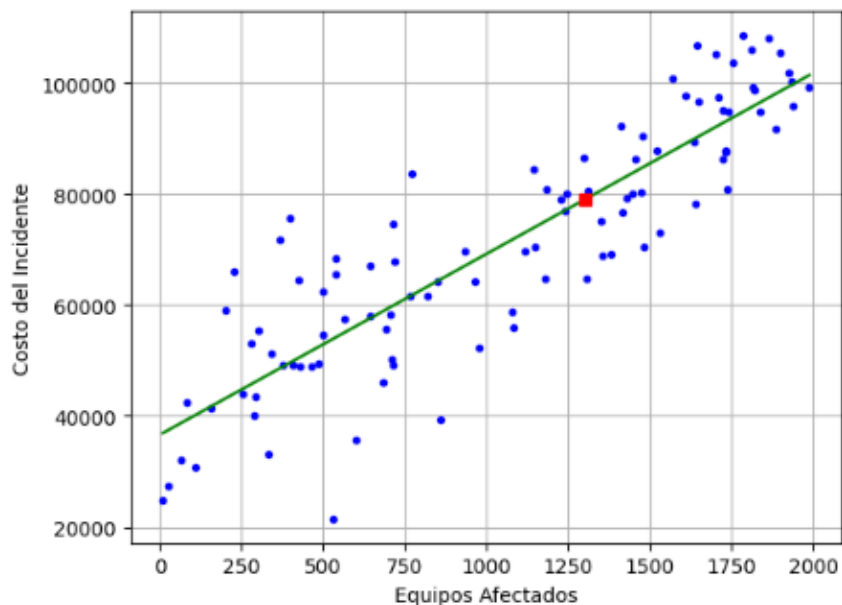


## 5.- Prediccion de nuevos ejemplos

```
[19]: x_new = np.array([[1300]]) # Numero de equipos afectados.  
      # Prediccion del costo que tendria el incidente.  
      Costo = lin_reg.predict(x_new)  
      print("El costo del incidente seria $", int(Costo[0]))
```

El costo del incidente seria \$ 79035

```
[20]: plt.plot(df['No_Equipos_Afectados'], df['Costo'], "b.")  
      plt.plot(X_min_max, y_train_pred, "g-")  
      plt.plot(x_new, Costo, "rs")  
      plt.xlabel("Equipos Afectados")  
      plt.ylabel("Costo del Incidente")  
      plt.grid(True)  
      plt.show()
```







#### IV. Conclusiones:

Este nuevo tema está enfocado principalmente en una problemática sobre ciberseguridad, a cerca de equipos afectados debido a un incidente de seguridad, la retroalimentación y el uso de librerías que anteriormente habíamos usado fue de mucha importancia y juega un papel fundamental que nos permite aplicar esos conocimientos en un ejercicio o caso de la vida real todo esto fue clave fundamental para poder facilitar este trabajo en el semestre presente, sinceramente esperamos seguir trabajando de esta manera con el docente ya que es una forma muy manejable y muy buena para entender y comprender las clases, cada uno de los temas que nos logra enseñar, algo ameno que como estudiantes hace que podamos comprender cada tema y así aprovechar cualquier momento para preguntar al maestro si en algún momento tenemos dudas sobre el tema.

El uso de las herramientas que nos proporciona el docente para trabajar facilitan la comprensión del tema en su extensión, mediante Anaconda Navigator y el uso de distintas librerías podemos analizar y comprender el movimiento sobre los datos que queramos analizar, todo eso con el fin de recopilar información útil para cada situación.

