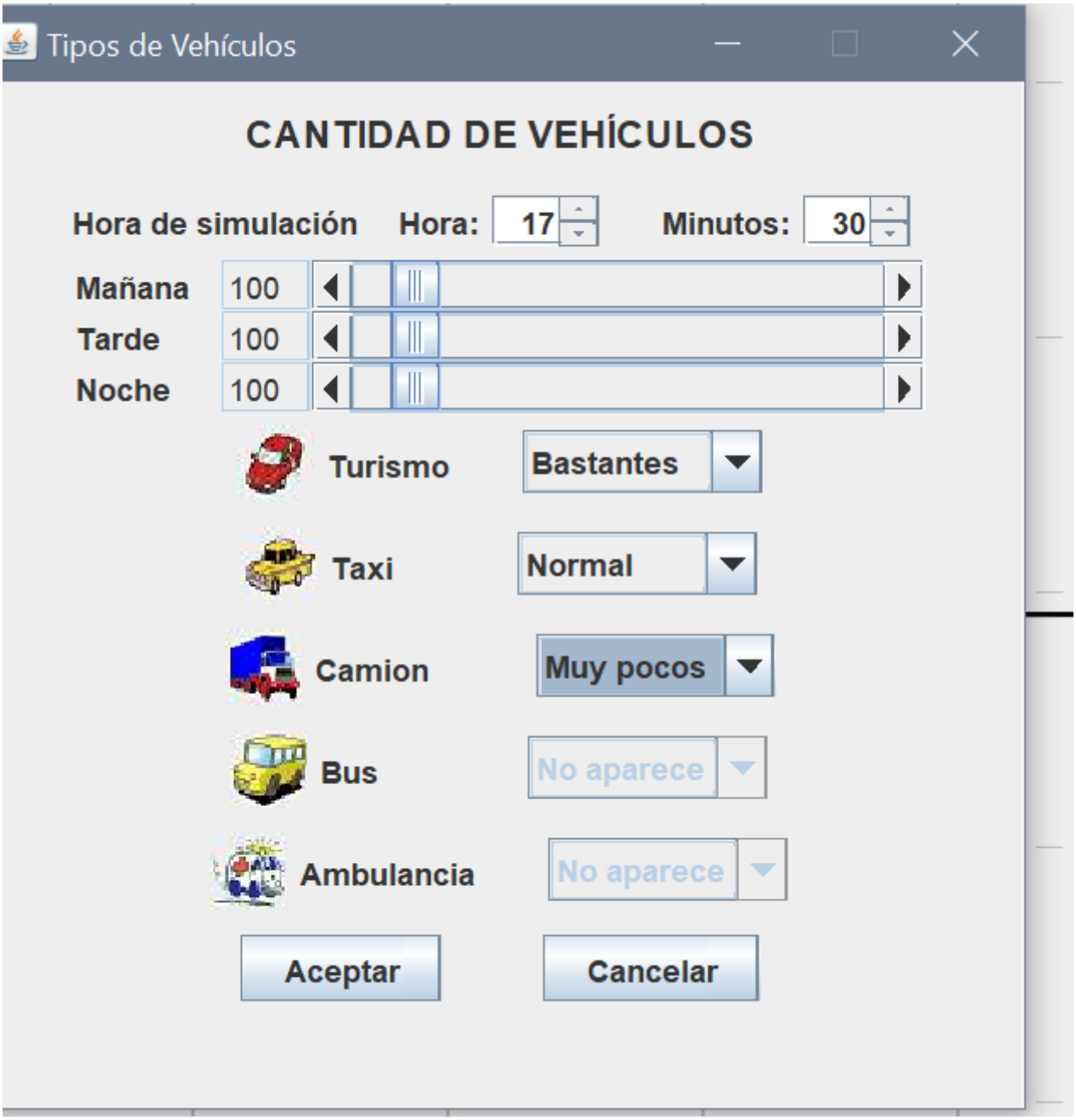
Universidad Politécnica Salesiana

Nombre: Angel Jadan ¶

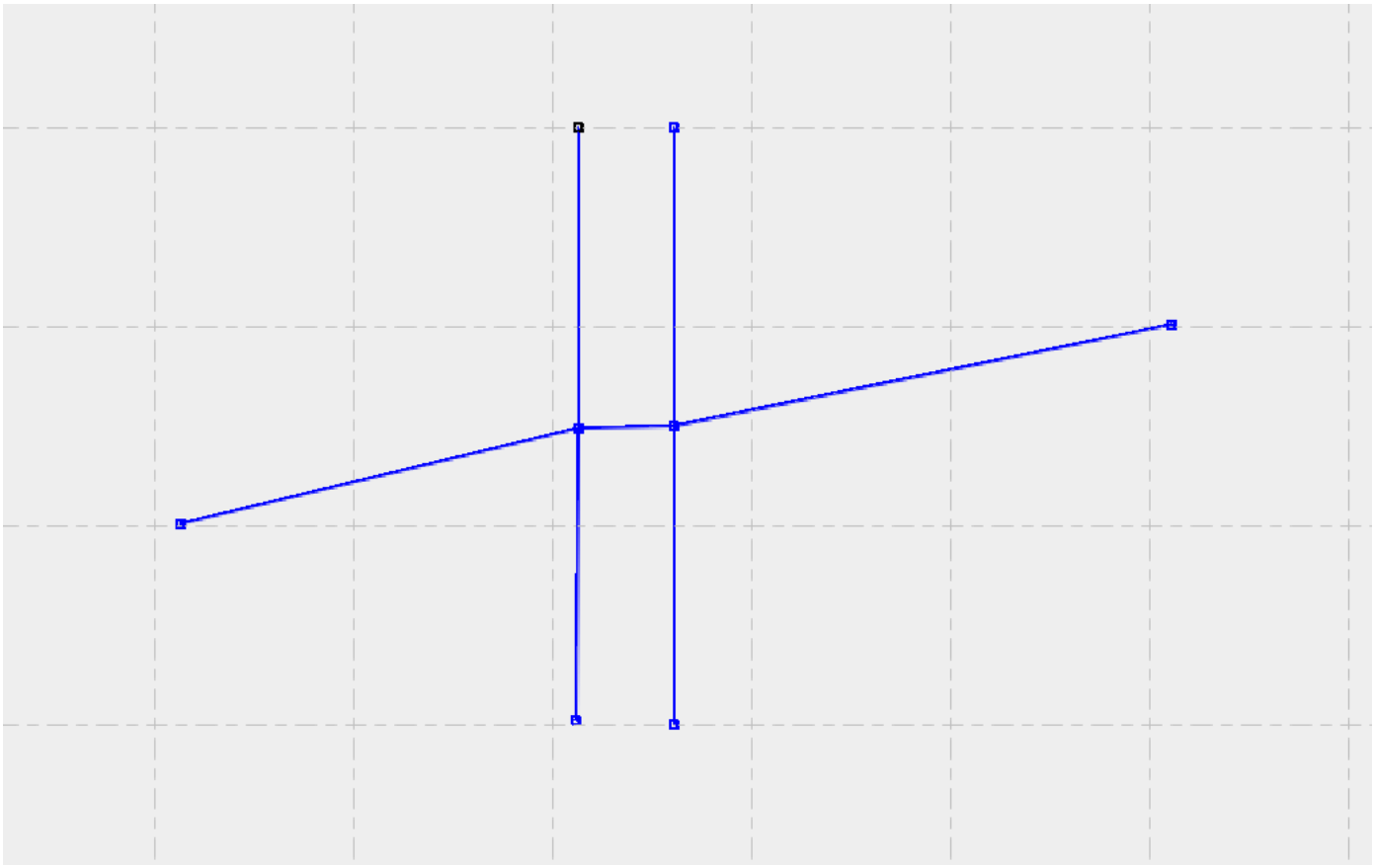
Fecha: 2/5/2021

Simulación de movimiento de vehiculos en la Ciudad de Cuenca, entre la Av. Loja y 10 de agosto.

Configuración de vehiculos para simulación.



Mapa de simulación de avenidas en SimTrafic.



Resultado de la simulación.

Fuente de datos: http://201.159.222.99/bitstream/datos/7226/1/13172.pdf (http://201.159.222.99/bitstream/datos/7226/1/13172.pdf)

In [99]:

```
import pandas as pd
import numpy as np
from datetime import datetime, timedelta
from sklearn.metrics import mean_squared_error
from scipy.optimize import curve_fit
from scipy.optimize import fsolve
from sklearn import linear_model
import matplotlib.pyplot as plt
%matplotlib inline
```

In [100]:

```
1 url = "datos.csv"
2 df = pd.read_csv(url,sep="\;")
3 df
```

```
<ipython-input-100-12075f068fde>:2: ParserWarning: Falling back to the 'pyth
on' engine because the 'c' engine does not support regex separators (separat
ors > 1 char and different from '\s+' are interpreted as regex); you can avo
id this warning by specifying engine='python'.
  df = pd.read_csv(url,sep="\;")
```

Out[100]:

	Name	Country Code	Indicator Name	Indicator Code	1960	1961	1962	
0	Aruba	ABW	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
1	Afganist n	AFG	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	4.132233	4.453443	4.878051	9.17
2	Angola	AGO	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
3	Albania	ALB	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
4	Andorra	AND	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
...	...	...	...	...	...	...	...	
259	Kosovo	XKX	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
260	Yemen, Rep. del	YEM	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
261	Sud frica	ZAF	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	29.550915	29.323968	29.406919	28.6
262	Zambia	ZMB	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	
263	Zimbabwe	ZWE	Exportaciones de bienes y servicios (% del PIB)	NE.EXP.GNFS.ZS	NaN	NaN	NaN	

264 rows × 64 columns

In [101]:

```
df = df[df['Name'].isin(['Ecuador'])]

df = df.loc[:,['Name','1960','1961','1962','1963','1964','1965','1966','1967','1968',\
            '1968','1969','1970','1971','1972','1973','1974','1975','1976',\
            '1977','1978','1979','1980','1981','1982','1983','1984','1985',\
            '1986','1987','1988','1989','1990','1991','1992','1993','1994',\
            '1995','1996','1997','1998','1999','2000','2001','2002','2003',\
            '2004','2005','2006','2007','2008','2009','2010','2011','2012',\
            '2013','2014','2015','2016','2017','2018','2019']]
df = df.set_index('Name').T

df
```

Out[101]:

Name	Ecuador
1960	9.547575
1961	8.957493
1962	10.241499
1963	9.233322
1964	8.900054
...	...
2015	21.258221
2016	19.504791
2017	20.832818
2018	22.604684
2019	23.390040

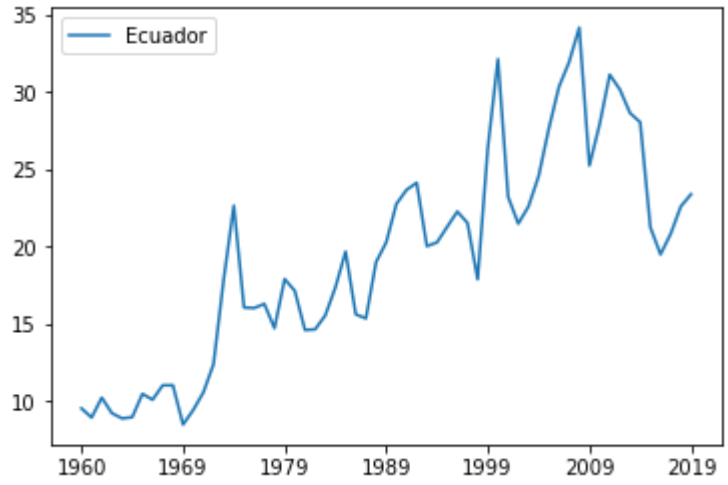
61 rows × 1 columns

In [102]:

```
df.plot(y="Ecuador")
```

Out[102]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1fdad31a2e0>
```



Y = mX + b

In [104]:

```
x = list(df.iloc [:, 0])
y = list(df.iloc [:, 0])

regr = linear_model.LinearRegression()

regr.fit(np.array(x).reshape(-1, 1), y)

print('Coeficintes: \n',regr.coef_)

print("Independiente termino: \n",regr.intercept_)
```

```
Coeficintes:
 [1.]
Independiente termino:
 3.552713678800501e-15
```

In [105]:

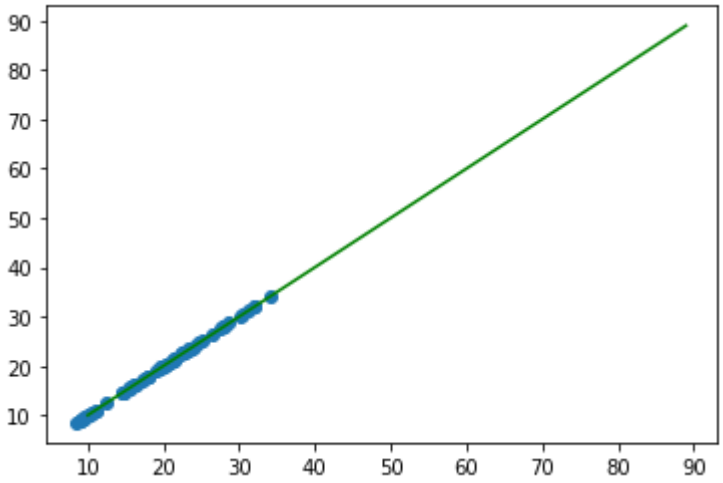
```
y_prediccion = regr.predict([[100]])
print(int(y_prediccion))
```

```
99
```

In [109]:

```
plt.scatter(x,y)
x_real = np.array(range(10,90))
print(x_real)
plt.plot(x_real, regr.predict(x_real.reshape(-1,1)), color='green')
plt.show()
```

```
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
 82 83 84 85 86 87 88 89]
```



Definamos la función en Python y realicemos elprocedimiento de ajuste de curva utilizado para el crecimiento logístico.

In [110]:

```
def modelo_logistico(x,a,b):
    return a+b*np.log(x)

exp_fit = curve_fit(modelo_logistico,x,y)
print(exp_fit)
```

```
(array([-30.98824426,  17.40266929]), array([[ 1.64810738, -0.56023698],
       [-0.56023698,  0.19386973]]))
```

In [111]:

```
pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
plt.rcParams['figure.figsize'] = [7, 7]
plt.rc('font', size=14)
# Real data
plt.scatter(x,y,label="Datos Reales",color="red")
# Predicted exponential curve
plt.plot(pred_x, [modelo_logistico(i,exp_fit[0][0],exp_fit[0][1]) for i in pred_x], lab
plt.legend()
plt.xlabel("Desde el 1 Enero 2020")
plt.ylabel("Total de personas infectadas")
plt.ylim((min(y)*0.9,max(y)*3.1)) # Definir los limites de Y
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-111-a9b5bfb0e5aa> in <module>
----> 1 pred_x = list(range(min(x),max(x)+50)) # Predecir 50 dias mas
      2 plt.rcParams['figure.figsize'] = [7, 7]
      3 plt.rc('font', size=14)
      4 # Real data
      5 plt.scatter(x,y,label="Datos Reales",color="red")

TypeError: 'float' object cannot be interpreted as an integer
```

In [ ]:

```
`
```