In [1]:
```
import pandas as pd
# Se importa la biblioteca pandas
```

In [2]:
```
pd.read_csv("PRODUCCION.csv")
# Muestra la tabla
```

Out[2]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
...	...	...	...	...	...	...	...	...
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

266 rows × 8 columns

In [3]:
```
Produccion_df=pd.read_csv("PRODUCCION.csv")
# Asignacion de la variable para el Dataframe
```

In [4]:
```
Produccion_df
# Listado del dataframe
```

Out[4]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
...	...	...	...	...	...	...	...	...
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

266 rows × 8 columns

In [5]:
```
type(Produccion_df)
# Se describe la estructura y el tipo del dataframe utilizado
```

Out[5]:
```
pandas.core.frame.DataFrame
```

In [6]:
```
Produccion_df.dtypes
# Muestra la estructura del dataframe y el tipo de elemento de cada campo
```

Out[6]:
```
Anio                          int64
Departamento                 object
Producto                     object
Area (ha)                    object
Produccion (ton)             object
Rendimiento (ha/ton)        float64
Produccion Nacional (ton)   float64
Area Nacional (ha)          float64
dtype: object
```

In [7]:
```
Produccion_df.columns
# Muesra las columnas del dataframe
```

Out[7]:
```
Index(['Anio', 'Departamento', 'Producto', 'Area (ha)', 'Produccion (ton)',
       'Rendimiento (ha/ton)', 'Produccion Nacional (ton)',
       'Area Nacional (ha)'],
      dtype='object')
```

In [8]:
```
Produccion_df.shape
# Muestra la cantidad de columnas y filas del dataframe
```

Out[8]:
```
(266, 8)
```

In [9]:
```
pd.unique(Produccion_df['Anio'])
# Muestra los valores del Anio y el tipo de manrea hizontal
```

Out[9]:
```
array([2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017,
       2018], dtype=int64)
```

In [10]:
```
pd.unique(Produccion_df['Departamento'])
# Muestra los valores del Depertamento y el tipo de manrea hizontal
```

Out[10]:
```
array(['ANTIOQUIA', 'BOLIVAR', 'BOYACA', 'CALDAS', 'CAQUETA', 'CASANARE',
       'CAUCA', 'CESAR', 'CHOCO', 'CUNDINAMARCA', 'HUILA', 'LA GUAJIRA',
       'MAGDALENA', 'META', 'NARIÑO', 'NORTE DE SANTANDER', 'PUTUMAYO',
       'QUINDIO', 'RISARALDA', 'SANTANDER', 'TOLIMA', 'VALLE DEL CAUCA',
       'ARAUCA', 'GUAVIARE'], dtype=object)
```

In [11]:
```
pd.unique(Produccion_df['Producto'])
# Muestra los valores del Producto y el tipo de manrea hizontal
```

Out[11]:
```
array(['CAFE'], dtype=object)
```

In [12]:
```
pd.unique(Produccion_df['Area (ha)'])
# Muestra los valores del Area(ha) y el tipo de manrea hizontal
```

Out[12]:
```
array(['112,343.60', '502.00', '11,374.50', '78,393.65', '2,295.00',
       '2,605.00', '53,471.00', '23,172.00', '290.00', '43,017.30',
       '89,661.56', '4,785.00', '17,506.00', '2,048.00', '24,458.50',
       '30,171.84', '35.00', '19,904.00', '47,689.25', '34,406.67',
       '91,679.10', '76,667.80', '114,694.00', '572.00', '10,778.50',
       '74,897.00', '2,735.00', '2,149.00', '56,208.00', '23,198.00',
       '90.00', '43,633.35', '89,131.20', '4,553.00', '17,521.00',
       '2,146.00', '25,582.00', '31.00', '19,571.00', '47,227.00',
       '34,169.37', '86,829.20', '72,419.00', '112,420.20', '770.00',
       '10,672.50', '73,083.00', '2,332.00', '1,904.00', '57,860.00',
       '23,420.00', '70.00', '43,475.84', '86,726.78', '4,488.00',
       '17,036.00', '2,216.00', '26,467.20', '33,552.58', '23.00',
       '19,052.00', '45,428.00', '37,985.90', '88,667.00', '67,001.30',
       '111,602.71', '0.00', '850.00', '9,427.00', '72,240.58',
       '2,536.00', '2,198.00', '55,162.00', '22,489.50', '157.50',
       '44,264.16', '87,139.53', '4,207.00', '17,000.00', '2,326.00',
       '23,504.05', '30,731.96', '24.00', '18,159.00', '47,308.00',
       '39,000.64', '84,658.70', '69,332.10', '106,419.57', '10.00',
       '8,441.74', '66,331.61', '2,810.00', '2,081.50', '54,246.42',
       '22,350.00', '37,478.87', '78,792.21', '4,100.00', '16,577.00',
       '2,578.00', '24,263.80', '21,520.45', '40.00', '20,139.30',
       '44,733.64', '37,282.04', '93,145.35', '68,038.40', '112,221.14',
       '870.00', '6,698.20', '54,871.88', '2,882.50', '2,322.00',
       '56,825.00', '22,911.00', '37,175.06', '79,809.34', '5,143.00',
       '17,686.00', '2,783.00', '27,806.40', '19,339.31', '42.00',
       '21,109.83', '45,588.03', '33,947.15', '90,904.48', '69,456.71',
       '109,755.50', '659.04', '9,289.05', '60,264.29', '2,905.84',
       '2,232.94', '74,105.64', '25,106.39', '125.01', '36,189.18',
       '118,200.88', '5,750.70', '17,016.72', '2,483.43', '32,136.51',
       '25,332.45', '24.27', '21,203.03', '39,615.60', '38,613.68',
       '97,308.81', '53,481.02', '110,115.86', '936.34', '9,834.39',
       '59,757.18', '3,074.92', '2,599.43', '77,068.46', '26,138.58',
       '136.88', '33,623.54', '128,273.15', '6,078.64', '18,533.11',
       '2,739.71', '33,608.32', '23,724.20', '101.16', '21,462.81',
       '40,154.46', '40,733.20', '100,832.91', '56,035.94', '109,649.61',
       '1,065.07', '10,461.85', '58,376.40', '3,410.56', '2,752.31',
       '77,405.83', '25,948.50', '137.47', '34,101.49', '130,452.40',
       '5,631.53', '17,996.31', '2,922.21', '33,490.93', '22,940.64',
       '128.65', '21,491.21', '41,732.03', '42,679.11', '103,368.73',
       '54,938.79', '105,666.60', '1,065.97', '10,181.80', '56,022.04',
       '3,392.22', '2,671.04', '78,421.95', '25,530.59', '134.96',
       '33,214.17', '126,052.15', '5,531.20', '17,745.80', '2,924.89',
       '32,750.16', '21,520.64', '20,041.70', '40,472.26', '41,387.79',
       '100,328.77', '52,648.25', '99,311.53', '1,137.42', '9,598.33',
       '51,854.59', '3,408.69', '2,436.63', '80,289.56', '25,158.80',
       '125.67', '30,894.16', '122,575.76', '5,340.80', '18,129.50',
       '2,926.85', '33,639.55', '21,409.77', '209.29', '17,699.67',
       '37,334.16', '42,327.26', '96,018.89', '51,470.86', '98,038.15',
       '1,182.13', '9,653.45', '50,762.22', '3,485.24', '2,360.55',
       '82,085.54', '23,915.45', '140.33', '29,085.24', '122,002.46',
       '4,810.97', '17,414.32', '2,761.01', '33,465.54', '20,873.04',
       '209.93', '16,374.73', '35,874.73', '42,269.07', '97,304.04',
       '48,305.31'], dtype=object)
```

In [13]:
```
pd.unique(Produccion_df['Produccion (ton)'])
# Muestra los valores del Produccion (Ton) y el tipo de manrea hizontal
```

Out[13]:
```
array(['120,500.80', '446.00', '9,683.10', '92,815.00', '2,134.00',
       '2,048.40', '51,348.00', '13,278.50', '205.90', '33,729.14',
       '129,052.51', '2,958.70', '14,005.00', '1,617.20', '31,770.05',
       '13,593.24', '34.00', '25,426.00', '72,842.55', '29,469.52',
       '112,322.38', '69,618.24', '113,505.20', '711.00', '9,547.30',
       '86,884.00', '2,469.00', '1,388.13', '48,073.00', '13,841.45',
       '68.00', '78,254.77', '131,316.47', '2,328.90', '14,017.00',
       '1,656.96', '31,262.50', '13,593.25', '35.60', '23,669.00',
       '60,079.00', '29,016.75', '101,201.88', '65,666.43', '103,703.00',
       '292.60', '8,567.97', '81,668.22', '2,332.00', '2,079.70',
       '47,221.00', '12,770.00', '78.75', '37,118.07', '104,609.42',
       '2,340.40', '13,412.80', '1,672.60', '27,487.71', '10,221.69',
       '26.70', '21,985.00', '53,648.00', '26,311.61', '88,633.10',
       '62,711.08', '121,253.38', '0.00', '510.00', '7,083.07',
       '95,957.90', '2,902.50', '2,564.86', '45,113.00', '13,276.08',
       '98.00', '37,214.80', '104,336.56', '2,393.00', '13,600.00',
       '2,221.90', '24,594.10', '22,111.65', '21,065.00', '72,091.00',
       '27,094.16', '94,230.20', '69,496.65', '115,267.98', '12.00',
       '5,643.39', '78,805.87', '2,528.40', '2,023.50', '41,645.39',
       '11,035.85', '32,780.35', '85,150.66', '1,933.00', '13,301.60',
       '2,533.75', '24,073.95', '12,332.00', '45.80', '20,814.11',
       '49,042.31', '22,089.82', '53,288.42', '65,475.63', '91,621.30',
       '652.50', '4,981.59', '54,115.96', '2,446.38', '1,718.25',
       '50,588.14', '19,994.35', '140.00', '30,786.41', '85,212.64',
       '3,434.30', '14,096.05', '2,133.10', '28,077.94', '12,214.54',
       '48.40', '18,030.13', '36,989.43', '23,271.89', '85,027.49',
       '61,190.55', '102,403.24', '395.07', '5,591.05', '58,634.19',
       '2,188.92', '1,338.56', '56,303.92', '15,050.27', '105.93',
       '24,993.74', '115,874.98', '3,447.31', '10,200.84', '1,650.41',
       '28,606.96', '15,185.79', '16.87', '20,599.27', '39,073.92',
       '30,227.02', '77,215.36', '42,948.40', '111,452.91', '606.93',
       '6,364.41', '62,869.38', '2,503.81', '1,688.60', '63,365.76',
       '16,935.63', '125.42', '25,118.55', '135,971.20', '3,923.80',
       '12,012.98', '1,950.84', '32,321.56', '15,108.55', '76.04',
       '22,518.42', '42,719.53', '34,512.79', '86,453.62', '49,799.28',
       '120,365.77', '1,089.74', '9,501.54', '67,231.37', '3,749.27',
       '2,626.73', '83,626.44', '22,240.81', '158.20', '31,165.15',
       '145,168.10', '4,317.50', '16,691.31', '3,206.35', '36,607.56',
       '20,267.64', '124.67', '24,694.56', '47,215.69', '47,304.16',
       '105,563.88', '57,583.56', '119,970.68', '1,128.32', '9,583.80',
       '66,661.14', '3,861.63', '2,638.88', '87,642.49', '22,649.03',
       '160.62', '31,413.34', '145,154.42', '4,387.19', '17,031.09',
       '3,322.42', '37,020.90', '19,590.10', '23,791.30', '47,357.02',
       '47,512.36', '105,976.19', '57,067.08', '140,398.62', '748.97',
       '7,638.99', '68,668.20', '5,108.33', '1,747.51', '97,922.49',
       '16,628.14', '158.85', '33,943.39', '133,787.95', '3,516.80',
       '11,937.90', '4,013.11', '35,004.18', '23,409.44', '282.18',
       '18,792.05', '46,779.71', '54,908.68', '94,556.71', '51,687.80',
       '141,898.91', '734.91', '7,780.34', '68,670.96', '5,280.40',
       '1,629.25', '102,147.00', '14,943.62', '181.42', '32,580.24',
       '136,161.86', '2,990.91', '10,826.24', '3,877.62', '35,679.42',
       '23,471.69', '289.50', '17,739.03', '45,918.75', '55,918.71',
       '97,451.31', '49,667.88'], dtype=object)
```

In [14]:
```
pd.unique(Produccion_df['Rendimiento (ha/ton)'])
# Muestra los valores del Rendimiento (ha/ton) y el tipo de manrea hizontal
```

Out[14]:
```
array([1.07, 0.89, 0.85, 1.18, 0.93, 0.79, 0.96, 0.57, 0.71, 0.78, 1.44,
       0.62, 0.8 , 1.3 , 0.45, 0.97, 1.28, 1.53, 0.86, 1.23, 0.91, 0.99,
       1.24, 1.16, 0.9 , 0.65, 0.6 , 0.76, 1.79, 1.47, 0.51, 0.77, 1.22,
       1.15, 1.21, 1.27, 1.17, 0.92, 0.38, 1.12, 1.  , 1.09, 0.82, 0.55,
       1.13, 0.52, 0.75, 1.04, 0.3 , 0.69, 0.94, 0.  , 1.33, 1.14, 0.59,
       0.84, 1.2 , 1.05, 0.72, 1.11, 1.52, 1.08, 0.67, 1.19, 0.49, 0.87,
       0.47, 0.98, 1.03, 1.1 , 0.74, 2.  , 0.83, 1.01, 0.63, 0.81, 0.88,
       0.66, 0.7 , 1.06, 0.64, 1.02, 0.95, 1.41, 1.32, 1.5 , 1.26, 1.37,
       1.35, 1.25, 1.45, 1.29, 1.4 , 1.38])
```

In [15]:
```
pd.unique(Produccion_df['Produccion Nacional (ton)'])
# Muestra los valores del Produccion nacional (ton) y el tipo de manrea hizontal
```

Out[15]:
```
array([1.454e+01, 5.000e-02, 1.170e+00, 1.120e+01, 2.600e-01, 2.500e-01,
       6.190e+00, 1.600e+00, 2.000e-02, 4.070e+00, 1.557e+01, 3.600e-01,
       1.690e+00, 2.000e-01, 3.830e+00, 1.640e+00, 0.000e+00, 3.070e+00,
       8.790e+00, 3.560e+00, 1.355e+01, 8.400e+00, 1.370e+01, 9.000e-02,
       1.150e+00, 1.049e+01, 3.000e-01, 1.700e-01, 5.800e+00, 1.670e+00,
       1.000e-02, 9.440e+00, 1.585e+01, 2.800e-01, 3.770e+00, 2.860e+00,
       7.250e+00, 3.500e+00, 1.221e+01, 7.930e+00, 1.463e+01, 4.000e-02,
       1.210e+00, 1.152e+01, 3.300e-01, 2.900e-01, 6.660e+00, 1.800e+00,
       5.240e+00, 1.476e+01, 1.890e+00, 2.400e-01, 3.880e+00, 1.440e+00,
       3.100e+00, 7.570e+00, 3.710e+00, 1.250e+01, 8.850e+00, 1.556e+01,
       7.000e-02, 9.100e-01, 1.231e+01, 3.700e-01, 5.790e+00, 1.700e+00,
       4.780e+00, 1.339e+01, 3.100e-01, 1.750e+00, 3.160e+00, 2.840e+00,
       2.700e+00, 9.250e+00, 3.480e+00, 1.209e+01, 8.920e+00, 1.800e+01,
       8.000e-02, 8.800e-01, 3.900e-01, 3.200e-01, 6.500e+00, 1.720e+00,
       5.120e+00, 1.330e+01, 2.080e+00, 4.000e-01, 3.760e+00, 1.930e+00,
       3.250e+00, 7.660e+00, 3.450e+00, 8.320e+00, 1.022e+01, 1.462e+01,
       1.000e-01, 7.900e-01, 8.630e+00, 2.700e-01, 8.070e+00, 3.190e+00,
       4.910e+00, 1.360e+01, 5.500e-01, 2.250e+00, 3.400e-01, 4.480e+00,
       1.950e+00, 2.880e+00, 5.900e+00, 1.357e+01, 9.760e+00, 1.570e+01,
       6.000e-02, 8.600e-01, 8.990e+00, 2.100e-01, 2.310e+00, 1.777e+01,
       5.300e-01, 1.560e+00, 4.390e+00, 2.330e+00, 5.990e+00, 4.640e+00,
       1.184e+01, 6.590e+00, 1.530e+01, 8.700e-01, 2.300e-01, 8.700e+00,
       1.867e+01, 5.400e-01, 1.650e+00, 4.440e+00, 2.070e+00, 3.090e+00,
       5.860e+00, 4.740e+00, 1.187e+01, 6.840e+00, 1.415e+01, 1.300e-01,
       1.120e+00, 7.900e+00, 4.400e-01, 9.830e+00, 2.620e+00, 3.660e+00,
       1.707e+01, 5.100e-01, 1.960e+00, 3.800e-01, 4.300e+00, 2.380e+00,
       2.900e+00, 5.550e+00, 5.560e+00, 1.241e+01, 6.770e+00, 1.405e+01,
       7.810e+00, 4.500e-01, 1.026e+01, 2.650e+00, 3.680e+00, 1.700e+01,
       1.990e+00, 4.340e+00, 2.290e+00, 2.790e+00, 6.680e+00, 1.649e+01,
       9.000e-01, 8.060e+00, 6.000e-01, 1.150e+01, 3.990e+00, 1.571e+01,
       4.100e-01, 1.400e+00, 4.700e-01, 4.110e+00, 2.750e+00, 3.000e-02,
       2.210e+00, 5.490e+00, 6.450e+00, 1.110e+01, 6.070e+00, 1.658e+01,
       8.020e+00, 6.200e-01, 1.900e-01, 1.194e+01, 3.810e+00, 1.591e+01,
       3.500e-01, 1.260e+00, 4.170e+00, 2.740e+00, 5.370e+00, 6.530e+00,
       1.139e+01])
```

In [16]:
```
pd.unique(Produccion_df['Area Nacional (ha)'])
# Muestra los valores del Area Nacional (ha) y el tipo de manrea hizontal
```

Out[16]:
```
array([1.466e+01, 7.000e-02, 1.480e+00, 1.023e+01, 3.000e-01, 3.400e-01,
       6.980e+00, 3.020e+00, 4.000e-02, 5.610e+00, 1.170e+01, 6.200e-01,
       2.280e+00, 2.700e-01, 3.190e+00, 3.940e+00, 0.000e+00, 2.600e+00,
       6.220e+00, 4.490e+00, 1.196e+01, 1.000e+01, 1.513e+01, 8.000e-02,
       1.420e+00, 9.880e+00, 3.600e-01, 2.800e-01, 7.410e+00, 3.060e+00,
       1.000e-02, 5.750e+00, 1.175e+01, 6.000e-01, 2.310e+00, 3.370e+00,
       3.980e+00, 2.580e+00, 6.230e+00, 4.510e+00, 1.145e+01, 9.550e+00,
       1.490e+01, 1.000e-01, 1.410e+00, 9.680e+00, 3.100e-01, 2.500e-01,
       7.670e+00, 3.100e+00, 5.760e+00, 1.149e+01, 5.900e-01, 2.260e+00,
       2.900e-01, 3.510e+00, 4.450e+00, 2.520e+00, 6.020e+00, 5.030e+00,
       8.880e+00, 1.499e+01, 1.100e-01, 1.270e+00, 9.710e+00, 2.000e-02,
       5.950e+00, 1.171e+01, 5.700e-01, 3.160e+00, 4.130e+00, 2.440e+00,
       6.360e+00, 5.240e+00, 1.137e+01, 9.310e+00, 1.494e+01, 1.200e-01,
       1.180e+00, 3.900e-01, 7.610e+00, 3.140e+00, 5.260e+00, 1.106e+01,
       5.800e-01, 2.330e+00, 3.410e+00, 2.830e+00, 6.280e+00, 5.230e+00,
       1.308e+01, 1.580e+01, 9.400e-01, 7.720e+00, 4.100e-01, 3.300e-01,
       8.000e+00, 3.220e+00, 1.123e+01, 7.200e-01, 2.490e+00, 3.910e+00,
       2.720e+00, 2.970e+00, 6.420e+00, 4.780e+00, 1.280e+01, 9.780e+00,
       1.422e+01, 9.000e-02, 1.200e+00, 7.810e+00, 3.800e-01, 9.600e+00,
       3.250e+00, 4.690e+00, 1.531e+01, 7.500e-01, 2.200e+00, 3.200e-01,
       4.160e+00, 3.280e+00, 2.750e+00, 5.130e+00, 5.000e+00, 1.261e+01,
       6.930e+00, 1.384e+01, 1.240e+00, 7.510e+00, 9.690e+00, 3.290e+00,
       4.230e+00, 1.612e+01, 7.600e-01, 4.220e+00, 2.980e+00, 2.700e+00,
       5.050e+00, 5.120e+00, 1.267e+01, 7.040e+00, 1.369e+01, 1.300e-01,
       1.310e+00, 7.290e+00, 4.300e-01, 9.660e+00, 3.240e+00, 4.260e+00,
       1.628e+01, 7.000e-01, 2.250e+00, 4.180e+00, 2.860e+00, 2.680e+00,
       5.210e+00, 5.330e+00, 1.290e+01, 6.860e+00, 1.359e+01, 1.400e-01,
       7.200e+00, 4.400e-01, 1.008e+01, 4.270e+00, 1.621e+01, 7.100e-01,
       4.210e+00, 2.770e+00, 5.200e+00, 5.320e+00, 6.770e+00, 1.318e+01,
       1.500e-01, 6.880e+00, 4.500e-01, 1.066e+01, 3.340e+00, 4.100e+00,
       1.627e+01, 2.410e+00, 4.470e+00, 2.840e+00, 3.000e-02, 2.350e+00,
       4.960e+00, 5.620e+00, 1.275e+01, 6.830e+00, 1.321e+01, 1.600e-01,
       1.300e+00, 6.840e+00, 4.700e-01, 3.920e+00, 1.643e+01, 6.500e-01,
       3.700e-01, 2.810e+00, 2.210e+00, 4.830e+00, 5.690e+00, 1.311e+01,
       6.510e+00])
```

In [17]:
```
Produccion_df['Anio']
# Muestra los valores del campo y el tipo de forma vertical
```

Out[17]:
```
0      2007
1      2007
2      2007
3      2007
4      2007
       ...
261    2018
262    2018
263    2018
264    2018
265    2018
Name: Anio, Length: 266, dtype: int64
```

In [18]:
```
Produccion_df['Departamento']
# Muestra los valores del departamento y tipo de forma vertical
```

Out[18]:
```
0            ANTIOQUIA
1              BOLIVAR
2               BOYACA
3               CALDAS
4              CAQUETA
             ...
261            QUINDIO
262          RISARALDA
263          SANTANDER
264             TOLIMA
265    VALLE DEL CAUCA
Name: Departamento, Length: 266, dtype: object
```

In [19]:
```
Produccion_df['Departamento']+ Produccion_df['Produccion (ton)']
# Muestra los valores de Departamento y el produccion con tipo
```

Out[19]:
```
0           ANTIOQUIA120,500.80
1                 BOLIVAR446.00
2                BOYACA9,683.10
3               CALDAS92,815.00
4               CAQUETA2,134.00
                   ...
261            QUINDIO17,739.03
262          RISARALDA45,918.75
263          SANTANDER55,918.71
264             TOLIMA97,451.31
265    VALLE DEL CAUCA49,667.88
Length: 266, dtype: object
```

In [20]:
```
Produccion_df['Produccion (ton)']+ Produccion_df['Producto']
# Muestra los valores de produccion y el Producto con tipo
```

Out[20]:
```
0      120,500.80CAFE
1          446.00CAFE
2        9,683.10CAFE
3       92,815.00CAFE
4        2,134.00CAFE
            ...
261     17,739.03CAFE
262     45,918.75CAFE
263     55,918.71CAFE
264     97,451.31CAFE
265     49,667.88CAFE
Length: 266, dtype: object
```

In [21]:
```
Produccion_df['Departamento'], Produccion_df['Produccion (ton)']
# Muestra los valores de Departamento y el produccion con tipo por separado
```

Out[21]:
```
(0            ANTIOQUIA
 1              BOLIVAR
 2               BOYACA
 3               CALDAS
 4              CAQUETA
               ...
 261            QUINDIO
 262          RISARALDA
 263          SANTANDER
 264             TOLIMA
 265    VALLE DEL CAUCA
 Name: Departamento, Length: 266, dtype: object,
 0      120,500.80
 1          446.00
 2        9,683.10
 3       92,815.00
 4        2,134.00
            ...
 261     17,739.03
 262     45,918.75
 263     55,918.71
 264     97,451.31
 265     49,667.88
 Name: Produccion (ton), Length: 266, dtype: object)
```

In [22]:
```
Produccion_df['Produccion (ton)'], Produccion_df['Producto']
# Muestra los valores de produccion y el Producto con tipo por separado
```

Out[22]:
```
(0      120,500.80
 1          446.00
 2        9,683.10
 3       92,815.00
 4        2,134.00
            ...
 261     17,739.03
 262     45,918.75
 263     55,918.71
 264     97,451.31
 265     49,667.88
 Name: Produccion (ton), Length: 266, dtype: object,
 0      CAFE
 1      CAFE
 2      CAFE
 3      CAFE
 4      CAFE
        ...
 261    CAFE
 262    CAFE
 263    CAFE
 264    CAFE
 265    CAFE
 Name: Producto, Length: 266, dtype: object)
```

In [23]:
```
Produccion_df.describe()
# Muestra los valores del dataframe de forma detallada y estadistica
```

Out[23]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
count	266.000000	266.000000	266.000000	266.000000
mean	2012.469925	0.936429	4.511316	4.511203
std	3.443484	0.267129	4.950568	4.565865
min	2007.000000	0.000000	0.000000	0.000000
25%	2010.000000	0.750000	0.352500	0.390000
50%	2012.000000	0.940000	2.720000	3.120000
75%	2015.000000	1.120000	7.147500	6.875000
max	2018.000000	2.000000	18.670000	16.430000

In [24]:
```
Produccion_df['Anio'].describe()
# Muestra los valores del anio de forma detallada y estadistica
```

Out[24]:
```
count     266.000000
mean     2012.469925
std         3.443484
min      2007.000000
25%      2010.000000
50%      2012.000000
75%      2015.000000
max      2018.000000
Name: Anio, dtype: float64
```

In [25]:
```
Produccion_df['Area (ha)'].describe()
# Muestra los valores del Area (ha) de forma detallada y estadistica
```

Out[25]:
```
count       266
unique      261
top       70.00
freq          2
Name: Area (ha), dtype: object
```

In [26]:
```
Produccion_df['Rendimiento (ha/ton)'].describe()
# Muestra los valores del Rendimiento (ha/ton) de forma detallada y estadistica
```

Out[26]:
```
count    266.000000
mean       0.936429
std        0.267129
min        0.000000
25%        0.750000
50%        0.940000
75%        1.120000
max        2.000000
Name: Rendimiento (ha/ton), dtype: float64
```

In [27]:
```
Produccion_df['Produccion Nacional (ton)'].describe()
# Muestra los valores del anio de forma detallada y estadistica
```

Out[27]:
```
count    266.000000
mean       4.511316
std        4.950568
min        0.000000
25%        0.352500
50%        2.720000
75%        7.147500
max       18.670000
Name: Produccion Nacional (ton), dtype: float64
```

In [28]:
```
Produccion_df['Anio'].min()
# Muestra el valor minimo de la columna
```

Out[28]: 2007

In [29]:
```
Produccion_df['Anio'].max()
# Muestra el valor maximo de la columna
```

Out[29]: 2018

In [30]:
```
Produccion_df['Anio'].count()
# Cuenta cuantos datos hay en cada columna
```

Out[30]: 266

In [31]:
```
Produccion_df['Anio'].mean()
# indica el promedio de cada
```

Out[31]: 2012.46992481203

In [32]:
```
Produccion_df['Rendimiento (ha/ton)'].min()
```

Out[32]: 0.0

In [33]:
```
Produccion_df['Rendimiento (ha/ton)'].max()
```

Out[33]: 2.0

In [34]:
```
Produccion_df['Rendimiento (ha/ton)'].count()
```

Out[34]: 266

In [35]:
```
Produccion_df['Rendimiento (ha/ton)'].mean()
```

Out[35]: 0.9364285714285712

In [36]:
```
Produccion_df['Produccion Nacional (ton)'].min()
```

Out[36]: 0.0

In [37]:
```
Produccion_df['Produccion Nacional (ton)'].max()
```

Out[37]: 18.67

In [38]:
```
Produccion_df['Produccion Nacional (ton)'].count()
```

Out[38]: 266

In [39]:
```
Produccion_df['Produccion Nacional (ton)'].mean()
```

Out[39]: 4.511315789473683

In [40]:
```
Produccion_df['Area Nacional (ha)'].min()
```

Out[40]: 0.0

In [41]:
```
Produccion_df['Area Nacional (ha)'].max()
```

Out[41]: 16.43

In [42]:
```
Produccion_df['Area Nacional (ha)'].count()
```

Out[42]: 266

In [43]:
```
Produccion_df['Area Nacional (ha)'].mean()
```

Out[43]: 4.511203007518795

In [44]:
```
Produccion_df.groupby('Anio')['Produccion (ton)'].count()[2012]
# Agrupa los datos del anio y cuenta las de producction que sean igual a 2012
```

Out[44]: 23

In [45]:
```
Produccion_df.groupby('Departamento')['Produccion (ton)'].count()['VALLE DEL CAUC
```

Out[45]: 12

In [46]:
```
Produccion_df.groupby('Anio')['Produccion (ton)'].count()
# Agrupa los datos del anio y muestra la cantidad de produccion (ton)
```

Out[46]:
```
Anio
2007    22
2008    22
2009    22
2010    23
2011    23
2012    23
2013    22
2014    22
2015    22
2016    21
2017    22
2018    22
Name: Produccion (ton), dtype: int64
```

In [47]:
```
Produccion_df.groupby('Departamento')['Produccion (ton)'].count()
```

Out[47]:
```
Departamento
ANTIOQUIA             12
ARAUCA                 2
BOLIVAR               12
BOYACA                12
CALDAS                12
CAQUETA               12
CASANARE              12
CAUCA                 12
CESAR                 12
CHOCO                 12
CUNDINAMARCA          12
GUAVIARE               1
HUILA                 12
LA GUAJIRA            12
MAGDALENA             12
META                  12
NARIÑO                12
NORTE DE SANTANDER    12
PUTUMAYO              11
QUINDIO               12
RISARALDA             12
SANTANDER             12
TOLIMA                12
VALLE DEL CAUCA       12
Name: Produccion (ton), dtype: int64
```

In [48]:
```
Produccion_df.groupby('Departamento')['Anio'].count()
```

Out[48]:
```
Departamento
ANTIOQUIA             12
ARAUCA                 2
BOLIVAR               12
BOYACA                12
CALDAS                12
CAQUETA               12
CASANARE              12
CAUCA                 12
CESAR                 12
CHOCO                 12
CUNDINAMARCA          12
GUAVIARE               1
HUILA                 12
LA GUAJIRA            12
MAGDALENA             12
META                  12
NARIÑO                12
NORTE DE SANTANDER    12
PUTUMAYO              11
QUINDIO               12
RISARALDA             12
SANTANDER             12
TOLIMA                12
VALLE DEL CAUCA       12
Name: Anio, dtype: int64
```

In [49]:
```
Producto_Counts=Produccion_df.groupby('Departamento')['Producto'].count()
print(Producto_Counts)
# Cuenta el departamento por la produccion y luego imprime los resultados
```

```
Departamento
ANTIOQUIA             12
ARAUCA                 2
BOLIVAR               12
BOYACA                12
CALDAS                12
CAQUETA               12
CASANARE              12
CAUCA                 12
CESAR                 12
CHOCO                 12
CUNDINAMARCA          12
GUAVIARE               1
HUILA                 12
LA GUAJIRA            12
MAGDALENA             12
META                  12
NARIÑO                12
NORTE DE SANTANDER    12
PUTUMAYO              11
QUINDIO               12
RISARALDA             12
SANTANDER             12
TOLIMA                12
VALLE DEL CAUCA       12
Name: Producto, dtype: int64
```

In [50]:
```
Produccion_df.head(20)
# Muestra las primeras 20 filas del dataframe
```

Out[50]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
5	2007	CASANARE	CAFE	2,605.00	2,048.40	0.79	0.25	0.34
6	2007	CAUCA	CAFE	53,471.00	51,348.00	0.96	6.19	6.98
7	2007	CESAR	CAFE	23,172.00	13,278.50	0.57	1.60	3.02
8	2007	CHOCO	CAFE	290.00	205.90	0.71	0.02	0.04
9	2007	CUNDINAMARCA	CAFE	43,017.30	33,729.14	0.78	4.07	5.61
10	2007	HUILA	CAFE	89,661.56	129,052.51	1.44	15.57	11.70
11	2007	LA GUAJIRA	CAFE	4,785.00	2,958.70	0.62	0.36	0.62
12	2007	MAGDALENA	CAFE	17,506.00	14,005.00	0.80	1.69	2.28
13	2007	META	CAFE	2,048.00	1,617.20	0.79	0.20	0.27
14	2007	NARIÑO	CAFE	24,458.50	31,770.05	1.30	3.83	3.19
15	2007	NORTE DE SANTANDER	CAFE	30,171.84	13,593.24	0.45	1.64	3.94
16	2007	PUTUMAYO	CAFE	35.00	34.00	0.97	0.00	0.00
17	2007	QUINDIO	CAFE	19,904.00	25,426.00	1.28	3.07	2.60
18	2007	RISARALDA	CAFE	47,689.25	72,842.55	1.53	8.79	6.22
19	2007	SANTANDER	CAFE	34,406.67	29,469.52	0.86	3.56	4.49

In [51]:
```
Produccion_df.tail(20)
# Muestra las ultimas 20 filas del dataframe
```

Out[51]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
246	2018	BOYACA	CAFE	9,653.45	7,780.34	0.81	0.91	1.30
247	2018	CALDAS	CAFE	50,762.22	68,670.96	1.35	8.02	6.84
248	2018	CAQUETA	CAFE	3,485.24	5,280.40	1.52	0.62	0.47
249	2018	CASANARE	CAFE	2,360.55	1,629.25	0.69	0.19	0.32
250	2018	CAUCA	CAFE	82,085.54	102,147.00	1.24	11.94	11.06
251	2018	CESAR	CAFE	23,915.45	14,943.62	0.62	1.75	3.22
252	2018	CHOCO	CAFE	140.33	181.42	1.29	0.02	0.02
253	2018	CUNDINAMARCA	CAFE	29,085.24	32,580.24	1.12	3.81	3.92
254	2018	HUILA	CAFE	122,002.46	136,161.86	1.12	15.91	16.43
255	2018	LA GUAJIRA	CAFE	4,810.97	2,990.91	0.62	0.35	0.65
256	2018	MAGDALENA	CAFE	17,414.32	10,826.24	0.62	1.26	2.35
257	2018	META	CAFE	2,761.01	3,877.62	1.40	0.45	0.37
258	2018	NARIÑO	CAFE	33,465.54	35,679.42	1.07	4.17	4.51
259	2018	NORTE DE SANTANDER	CAFE	20,873.04	23,471.69	1.12	2.74	2.81
260	2018	PUTUMAYO	CAFE	209.93	289.50	1.38	0.03	0.03
261	2018	QUINDIO	CAFE	16,374.73	17,739.03	1.08	2.07	2.21
262	2018	RISARALDA	CAFE	35,874.73	45,918.75	1.28	5.37	4.83
263	2018	SANTANDER	CAFE	42,269.07	55,918.71	1.32	6.53	5.69
264	2018	TOLIMA	CAFE	97,304.04	97,451.31	1.00	11.39	13.11
265	2018	VALLE DEL CAUCA	CAFE	48,305.31	49,667.88	1.03	5.80	6.51

In [52]:
```
Produccion_df['Produccion Nacional (ton)'].std()
# Describe la desviacion estandar
```

Out[52]:
```
4.950567735489969
```

In [53]:
```
Anio_grouped=Produccion_df.groupby('Anio')
print(Anio_grouped)
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x0000000008454848>
```

In [54]:
```
Anio_grouped.describe()
```

Out[54]:

	Rendimiento (ha/ton)								Produccion Nacional (t			
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	r
Anio													
2007	22.0	0.950455	0.279566	0.45	0.7900	0.900	1.1525	1.53	22.0	4.545909	...	7.8475	15
2008	22.0	0.982727	0.322670	0.45	0.7775	0.905	1.2000	1.79	22.0	4.545455	...	7.7600	15
2009	22.0	0.881364	0.264652	0.30	0.7600	0.930	1.1125	1.21	22.0	4.545455	...	7.3425	14
2010	23.0	0.906087	0.324692	0.00	0.7050	0.960	1.1250	1.52	23.0	4.348261	...	7.3550	15
2011	23.0	0.854348	0.238305	0.47	0.6100	0.900	1.0550	1.20	23.0	4.348696	...	7.0800	18
2012	23.0	0.858696	0.329618	0.00	0.7450	0.830	0.9150	2.00	23.0	4.347391	...	6.9850	14
2013	22.0	0.759545	0.145421	0.60	0.6000	0.755	0.8800	0.99	22.0	4.545455	...	6.4400	17
2014	22.0	0.822273	0.157629	0.64	0.6500	0.815	0.9500	1.06	22.0	4.545455	...	6.5950	18
2015	22.0	1.024545	0.110096	0.77	0.9350	1.065	1.1075	1.15	22.0	4.544545	...	6.4675	17
2016	21.0	1.063810	0.116725	0.79	0.9600	1.120	1.1500	1.19	21.0	4.761429	...	6.6800	17
2017	22.0	1.068182	0.272443	0.66	0.8450	1.090	1.2900	1.50	22.0	4.545909	...	6.3550	16
2018	22.0	1.079545	0.296672	0.62	0.8575	1.120	1.3125	1.52	22.0	4.545455	...	6.3475	16

12 rows × 24 columns

In [55]:
```
Anio_grouped.mean()
```

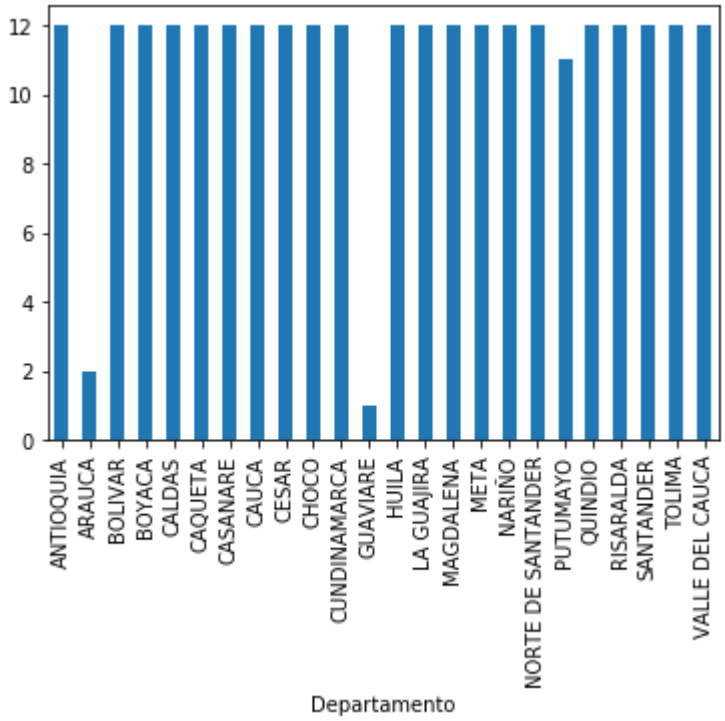
Out[55]:

	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio			
2007	0.950455	4.545909	4.545455
2008	0.982727	4.545455	4.545000
2009	0.881364	4.545455	4.544545
2010	0.906087	4.348261	4.347826
2011	0.854348	4.348696	4.347826
2012	0.858696	4.347391	4.347826
2013	0.759545	4.545455	4.545000
2014	0.822273	4.545455	4.545455
2015	1.024545	4.544545	4.545455
2016	1.063810	4.761429	4.761905
2017	1.068182	4.545909	4.545455
2018	1.079545	4.545455	4.546364

In [56]:
```
%matplotlib inline
# Se carga la libreria para graficos de pandas denominada matplotlib
```

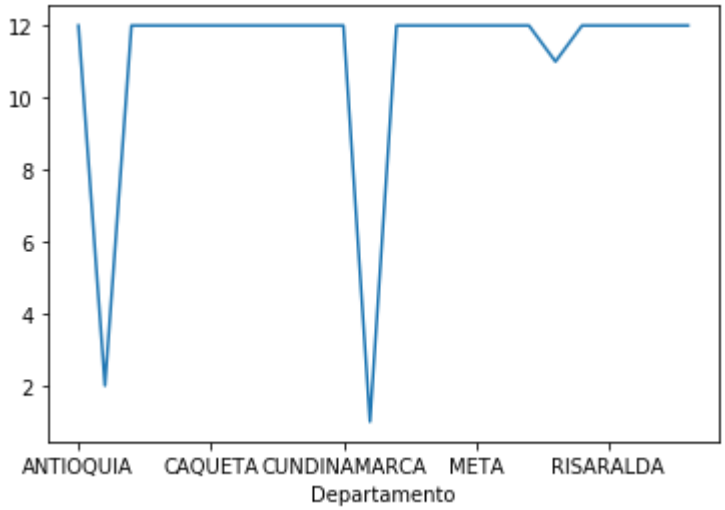
In [57]:
```
Producto_Counts.plot(kind='bar')
```

Out[57]: <matplotlib.axes._subplots.AxesSubplot at 0x950e988>



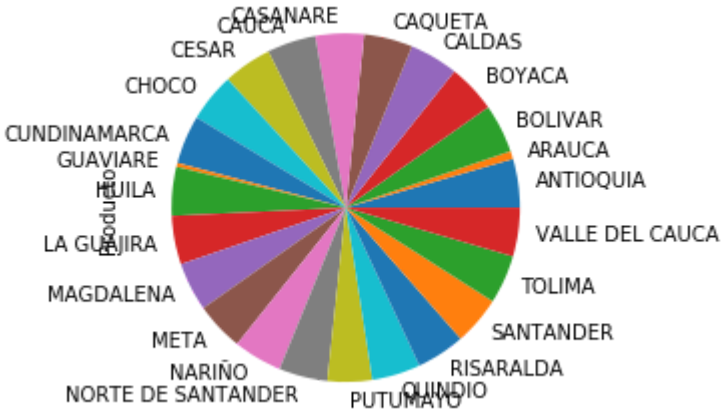
In [58]:
```
Producto_Counts.plot(kind='line')
```

Out[58]: <matplotlib.axes._subplots.AxesSubplot at 0x4125b08>



In [59]:
```
Producto_Counts.plot(kind='pie')
```

Out[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x9672a08>
```



In [60]:
```
Produccion_df.index
# Muestra la cantidad de datos por columnas
```

Out[60]:
```
RangeIndex(start=0, stop=266, step=1)
```

In [61]:
```
Produccion_df.size
# Muestra el numero de elementos del dataframe
```

Out[61]:
```
2128
```

In [62]:
```
Produccion_df.info
#Devuelve informacion sobre el dataframe
```

Out[62]:
```
<bound method DataFrame.info of      Anio     Departamento Producto   Area (ha)
Produccion (ton)  \
0    2007        ANTIOQUIA     CAFE  112,343.60       120,500.80
1    2007          BOLIVAR     CAFE      502.00           446.00
2    2007           BOYACA     CAFE   11,374.50         9,683.10
3    2007           CALDAS     CAFE   78,393.65        92,815.00
4    2007          CAQUETA     CAFE    2,295.00         2,134.00
..    ...              ...      ...         ...              ...
261  2018          QUINDIO     CAFE   16,374.73        17,739.03
262  2018        RISARALDA     CAFE   35,874.73        45,918.75
263  2018        SANTANDER     CAFE   42,269.07        55,918.71
264  2018           TOLIMA     CAFE   97,304.04        97,451.31
265  2018  VALLE DEL CAUCA     CAFE   48,305.31        49,667.88

     Rendimiento (ha/ton)  Produccion Nacional (ton)  Area Nacional (ha)
0                    1.07                      14.54               14.66
1                    0.89                       0.05                0.07
2                    0.85                       1.17                1.48
3                    1.18                      11.20               10.23
4                    0.93                       0.26                0.30
..                    ...                        ...                 ...
261                  1.08                       2.07                2.21
262                  1.28                       5.37                4.83
263                  1.32                       6.53                5.69
264                  1.00                      11.39               13.11
265                  1.03                       5.80                6.51

[266 rows x 8 columns]>
```

In [63]:
```
Produccion_df.iloc[1,:8]
# Devuelve un dataframe con los elementos de las filas de la lista y de las colum
```

Out[63]:
```
Anio                          2007
Departamento                BOLIVAR
Producto                       CAFE
Area (ha)                    502.00
Produccion (ton)             446.00
Rendimiento (ha/ton)           0.89
Produccion Nacional (ton)      0.05
Area Nacional (ha)             0.07
Name: 1, dtype: object
```

In [64]:
```
Produccion_df.loc[:31,'Departamento']
# Devuelve un dataframe con los elementos que se encuentra en la fila 32 con los
```

Out[64]:
```
0              ANTIOQUIA
1                BOLIVAR
2                 BOYACA
3                 CALDAS
4                CAQUETA
5               CASANARE
6                  CAUCA
7                  CESAR
8                  CHOCO
9           CUNDINAMARCA
10                 HUILA
11            LA GUAJIRA
12             MAGDALENA
13                  META
14               NARIÑO
15    NORTE DE SANTANDER
16              PUTUMAYO
17              QUINDIO
18             RISARALDA
19             SANTANDER
20                TOLIMA
21       VALLE DEL CAUCA
22             ANTIOQUIA
23               BOLIVAR
24                BOYACA
25                CALDAS
26               CAQUETA
27              CASANARE
28                 CAUCA
29                 CESAR
30                 CHOCO
31          CUNDINAMARCA
Name: Departamento, dtype: object
```

In [65]:
```
Produccion_df.isnull().sum()
# Muestra si en el datrame hay un valor vacio
```

Out[65]:
```
Anio                          0
Departamento                  0
Producto                      0
Area (ha)                     0
Produccion (ton)              0
Rendimiento (ha/ton)          0
Produccion Nacional (ton)     0
Area Nacional (ha)            0
dtype: int64
```

In [66]:
```
Produccion_df.duplicated().sum()
# Muestra si el dataframe presenta un valor duplicado
```

Out[66]:
```
0
```

In [110]:
```
Produccion_df[1:20]
# Muestra los valores de la fila 1 hasta la fila 20
```

Out[110]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30
5	2007	CASANARE	CAFE	2,605.00	2,048.40	0.79	0.25	0.34
6	2007	CAUCA	CAFE	53,471.00	51,348.00	0.96	6.19	6.98
7	2007	CESAR	CAFE	23,172.00	13,278.50	0.57	1.60	3.02
8	2007	CHOCO	CAFE	290.00	205.90	0.71	0.02	0.04
9	2007	CUNDINAMARCA	CAFE	43,017.30	33,729.14	0.78	4.07	5.61
10	2007	HUILA	CAFE	89,661.56	129,052.51	1.44	15.57	11.70
11	2007	LA GUAJIRA	CAFE	4,785.00	2,958.70	0.62	0.36	0.62
12	2007	MAGDALENA	CAFE	17,506.00	14,005.00	0.80	1.69	2.28
13	2007	META	CAFE	2,048.00	1,617.20	0.79	0.20	0.27
14	2007	NARIÑO	CAFE	24,458.50	31,770.05	1.30	3.83	3.19
15	2007	NORTE DE SANTANDER	CAFE	30,171.84	13,593.24	0.45	1.64	3.94
16	2007	PUTUMAYO	CAFE	35.00	34.00	0.97	0.00	0.00
17	2007	QUINDIO	CAFE	19,904.00	25,426.00	1.28	3.07	2.60
18	2007	RISARALDA	CAFE	47,689.25	72,842.55	1.53	8.79	6.22
19	2007	SANTANDER	CAFE	34,406.67	29,469.52	0.86	3.56	4.49

In [68]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn  as sns
from pandas_profiling import ProfileReport
from pathlib import Path
```

In [69]:
```
dir_prin=Path.cwd()
dir_entrada= dir_prin/'Produccion_csv'
df=pd.read_csv('Produccion.csv')
```

In [70]:
```
Produccion_df.head()
```

Out[70]:

	Anio	Departamento	Producto	Area (ha)	Produccion (ton)	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
0	2007	ANTIOQUIA	CAFE	112,343.60	120,500.80	1.07	14.54	14.66
1	2007	BOLIVAR	CAFE	502.00	446.00	0.89	0.05	0.07
2	2007	BOYACA	CAFE	11,374.50	9,683.10	0.85	1.17	1.48
3	2007	CALDAS	CAFE	78,393.65	92,815.00	1.18	11.20	10.23
4	2007	CAQUETA	CAFE	2,295.00	2,134.00	0.93	0.26	0.30

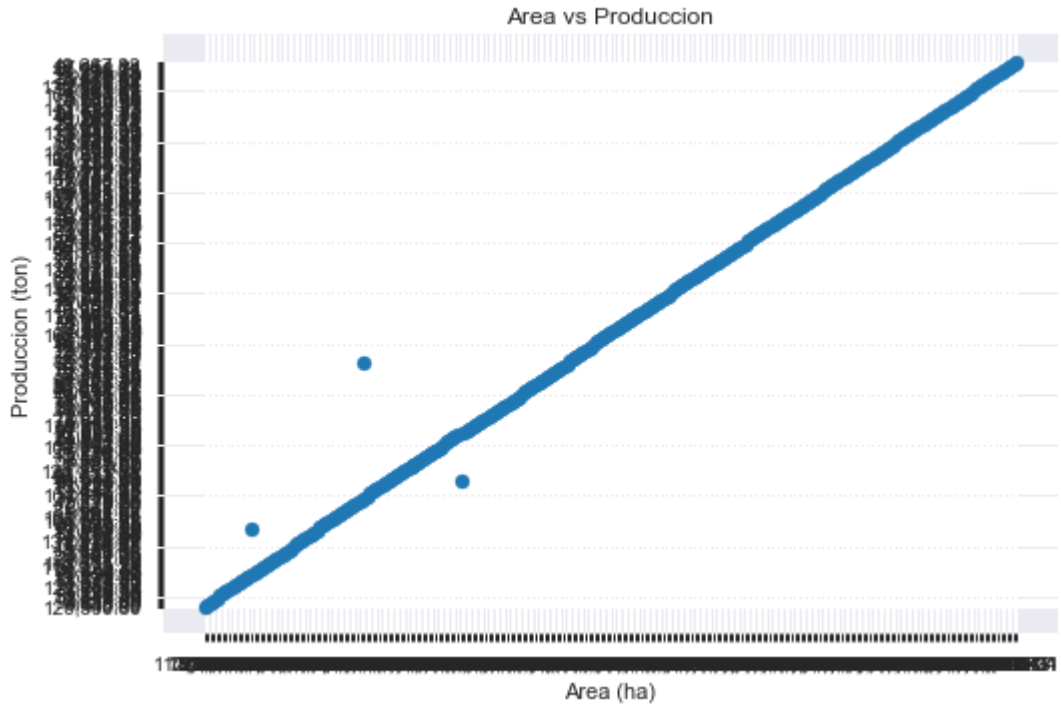
In [71]:
```
# estas instrucciones aun no las voy a emplear, por eso están con el simbolo #
#arreglo=list(produccion_df.columns)
#produccion1_df= produccion_df[arreglo[2:len(arreglo)]]
#produccion1_df
#print(produccion1_df)
#arreglo2.describe()
```

In [72]:
```
Produccion_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 266 entries, 0 to 265
Data columns (total 8 columns):
 #   Column                     Non-Null Count  Dtype  
---  ------                     --------------  -----  
 0   Anio                       266 non-null    int64  
 1   Departamento               266 non-null    object 
 2   Producto                   266 non-null    object 
 3   Area (ha)                  266 non-null    object 
 4   Produccion (ton)           266 non-null    object 
 5   Rendimiento (ha/ton)       266 non-null    float64
 6   Produccion Nacional (ton)  266 non-null    float64
 7   Area Nacional (ha)         266 non-null    float64
dtypes: float64(3), int64(1), object(4)
memory usage: 16.8+ KB
```

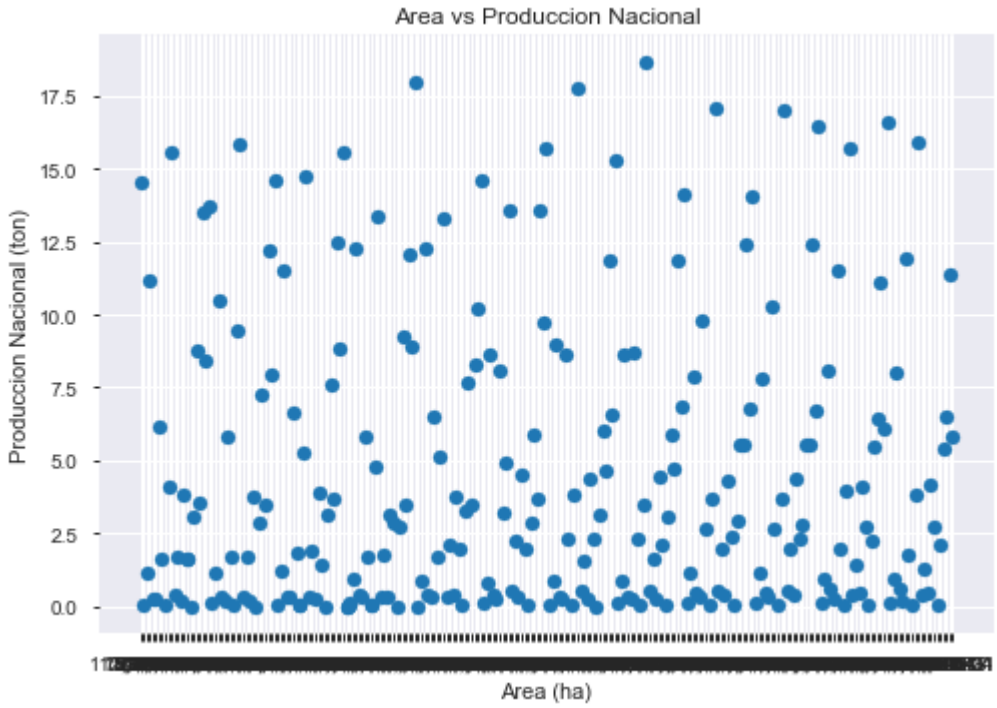
In [73]:
```
plt.scatter(Produccion_df['Area (ha)'],Produccion_df['Produccion (ton)'])
plt.title('Area vs Produccion')
plt.xlabel('Area (ha)')
plt.ylabel("Produccion (ton)")
```

Out[73]:
```
Text(0, 0.5, 'Produccion (ton)')
```



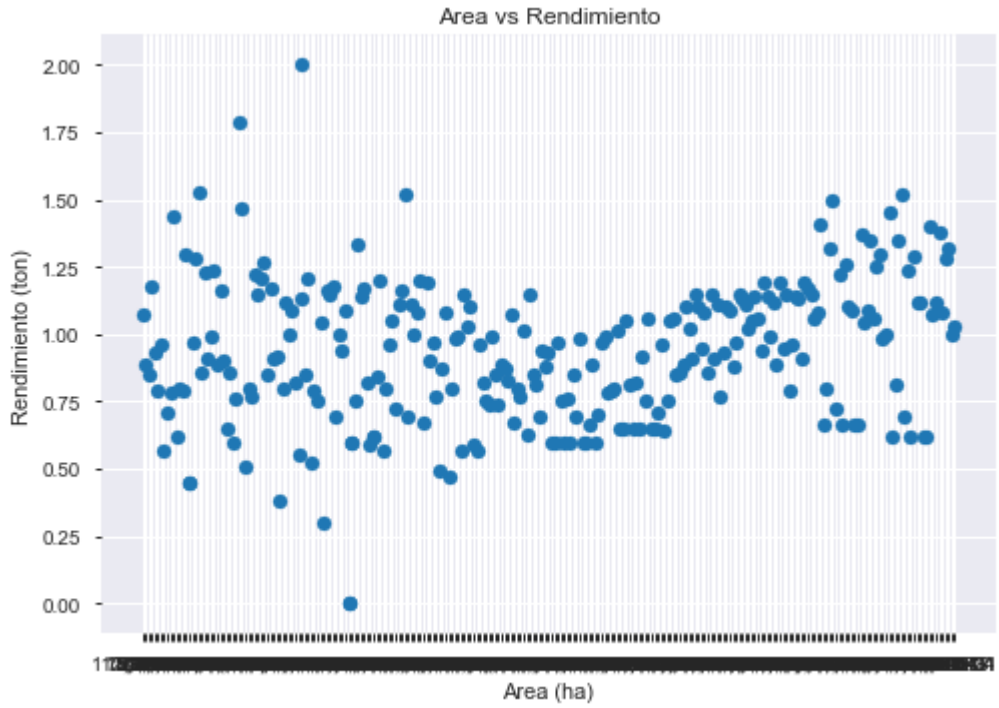
In [74]:
```
plt.scatter(Produccion_df['Area (ha)'],Produccion_df['Produccion Nacional (ton)']
plt.title('Area vs Produccion Nacional')
plt.xlabel('Area (ha)')
plt.ylabel("Produccion Nacional (ton)")
```

Out[74]:
```
Text(0, 0.5, 'Produccion Nacional (ton)')
```



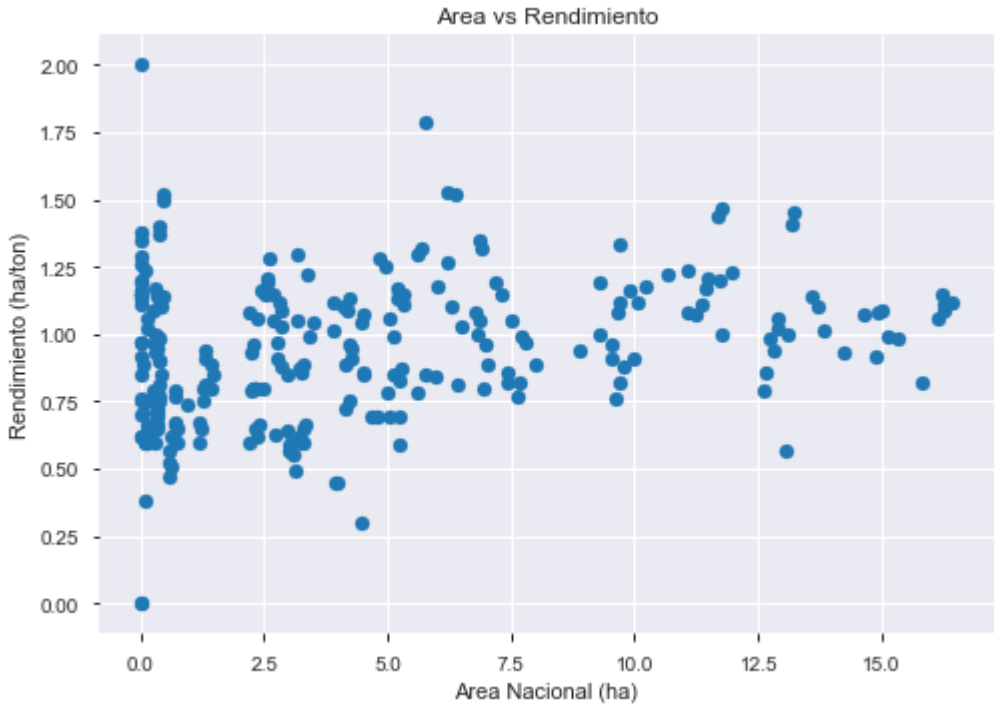
In [75]:
```
plt.scatter(Produccion_df['Area (ha)'],Produccion_df['Rendimiento (ha/ton)'])
plt.title('Area vs Rendimiento')
plt.xlabel('Area (ha)')
plt.ylabel("Rendimiento (ton)")
```

Out[75]:
```
Text(0, 0.5, 'Rendimiento (ton)')
```



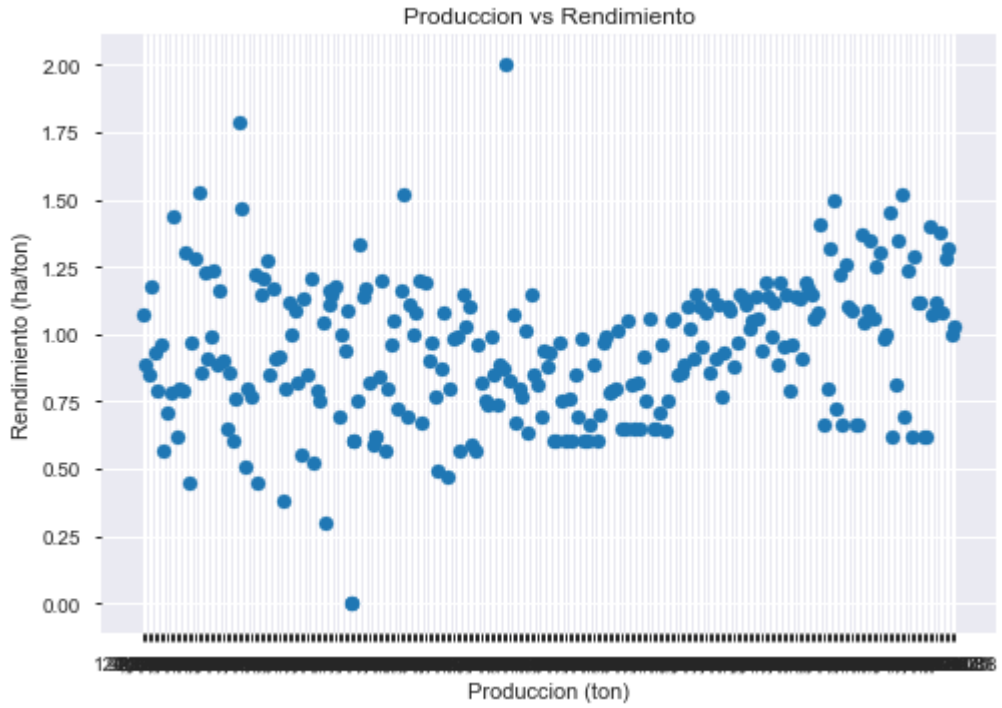
In [76]:
```
plt.scatter(Produccion_df['Area Nacional (ha)'],Produccion_df['Rendimiento (ha/to
plt.title('Area vs Rendimiento')
plt.xlabel('Area Nacional (ha)')
plt.ylabel("Rendimiento (ha/ton)")
```

Out[76]:
```
Text(0, 0.5, 'Rendimiento (ha/ton)')
```



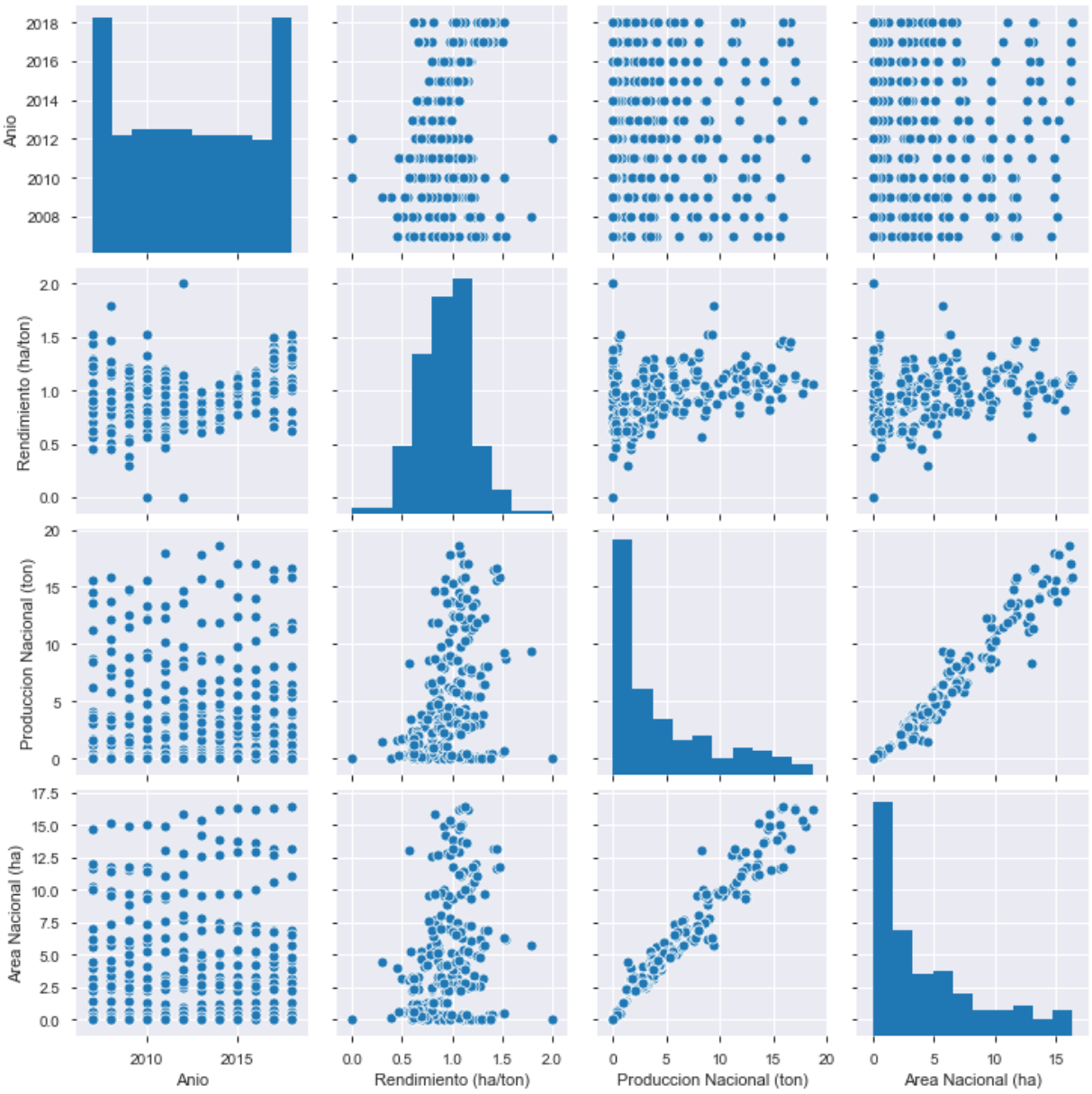
In [77]:
```
plt.scatter(Produccion_df['Produccion (ton)'],Produccion_df['Rendimiento (ha/ton)
plt.title('Produccion vs Rendimiento')
plt.xlabel('Produccion (ton)')
plt.ylabel("Rendimiento (ha/ton)")
```

Out[77]:
```
Text(0, 0.5, 'Rendimiento (ha/ton)')
```



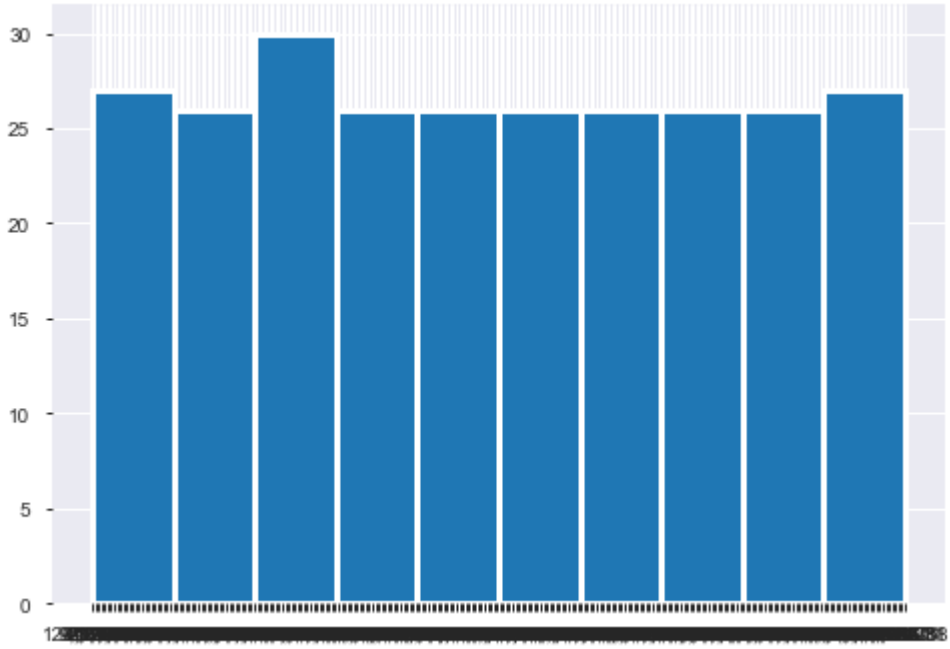
In [78]:
```
sns.pairplot(Produccion_df)
```

Out[78]:
```
<seaborn.axisgrid.PairGrid at 0xf08a6c8>
```



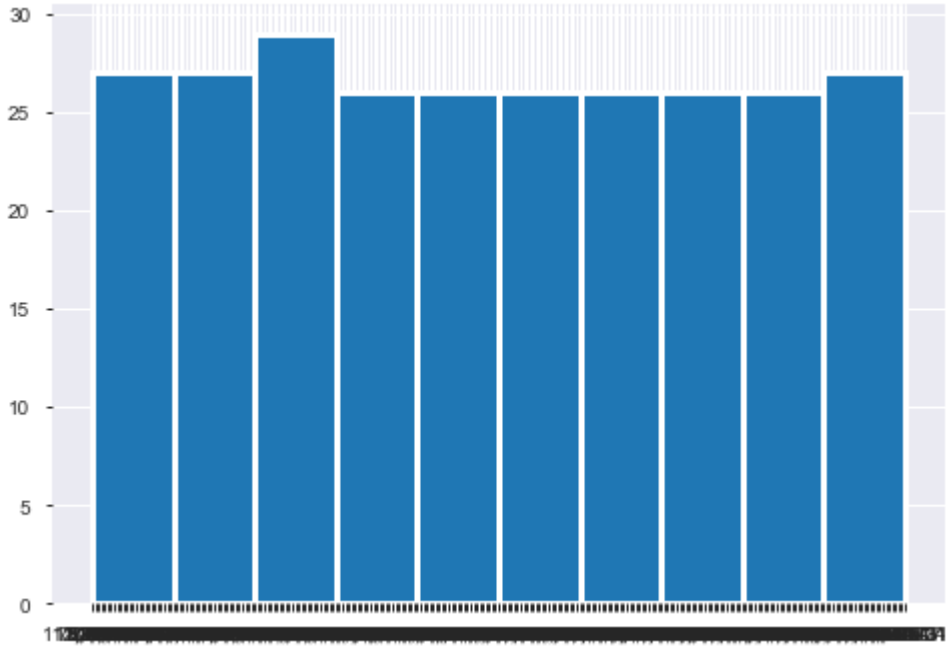
In [79]:
```
plt.hist(Produccion_df['Produccion (ton)'], edgecolor='white', linewidth=3)
# Histograma de Produccion (ton)
```

Out[79]:
```
(array([27., 26., 30., 26., 26., 26., 26., 26., 26., 27.]),
 array([  0. ,  26.1,  52.2,  78.3, 104.4, 130.5, 156.6, 182.7, 208.8,
        234.9, 261. ]),
 <a list of 10 Patch objects>)
```



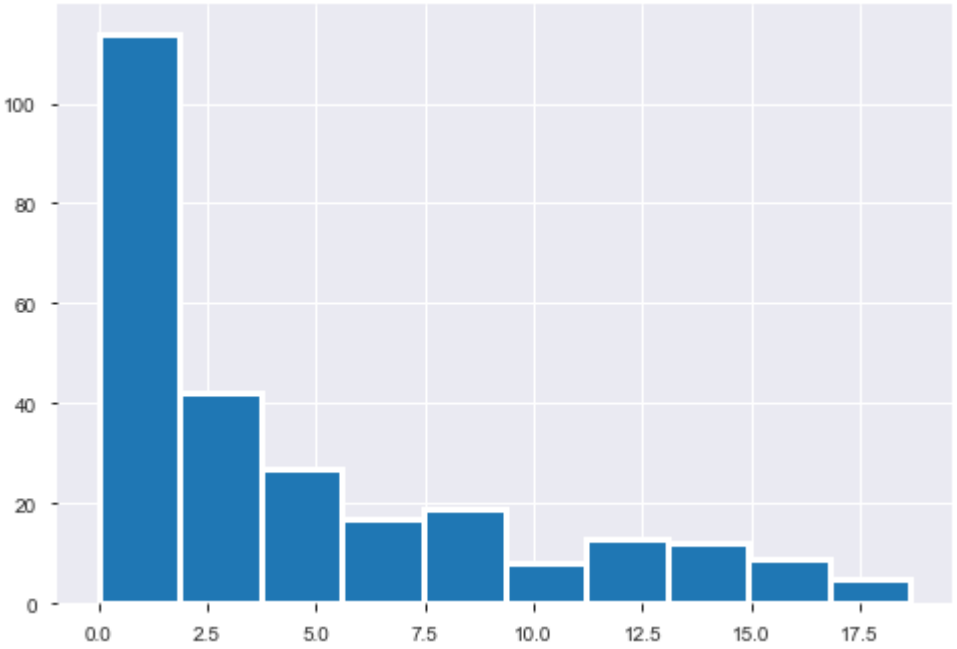
In [80]:
```
plt.hist(Produccion_df['Area (ha)'], edgecolor='white', linewidth=3)
```

Out[80]:
```
(array([27., 27., 29., 26., 26., 26., 26., 26., 26., 27.]),
 array([  0.,  26.,  52.,  78., 104., 130., 156., 182., 208., 234., 260.]),
 <a list of 10 Patch objects>)
```



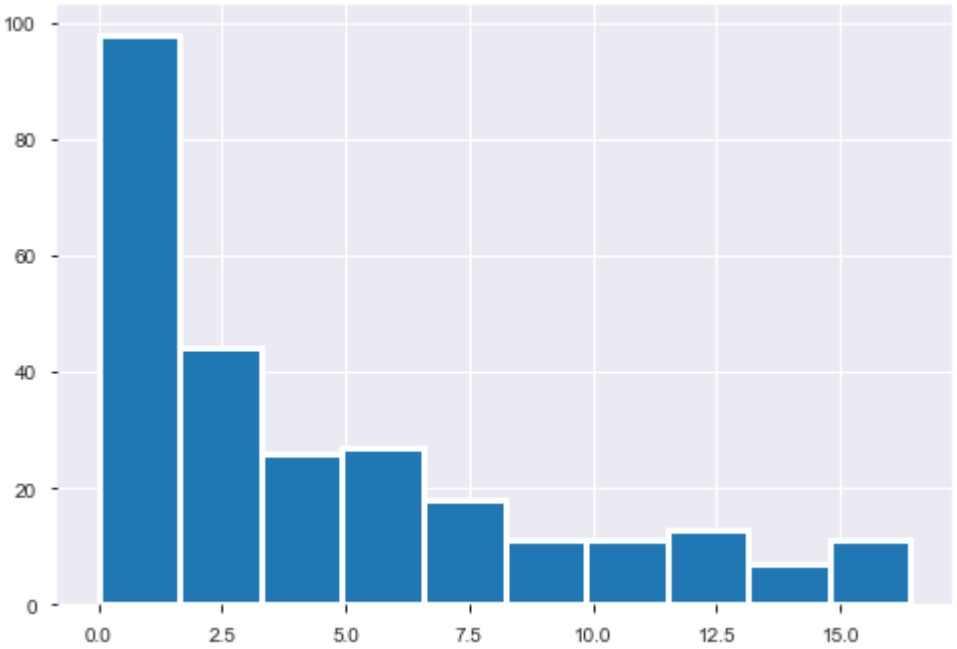
In [81]:
```
plt.hist(Produccion_df['Produccion Nacional (ton)'], edgecolor='white', linewidth
# Histograma de la Produccion Nacional
```

Out[81]:
```
(array([114.,  42.,  27.,  17.,  19.,   8.,  13.,  12.,   9.,   5.]),
 array([ 0.   ,  1.867,  3.734,  5.601,  7.468,  9.335, 11.202, 13.069,
        14.936, 16.803, 18.67 ]),
 <a list of 10 Patch objects>)
```



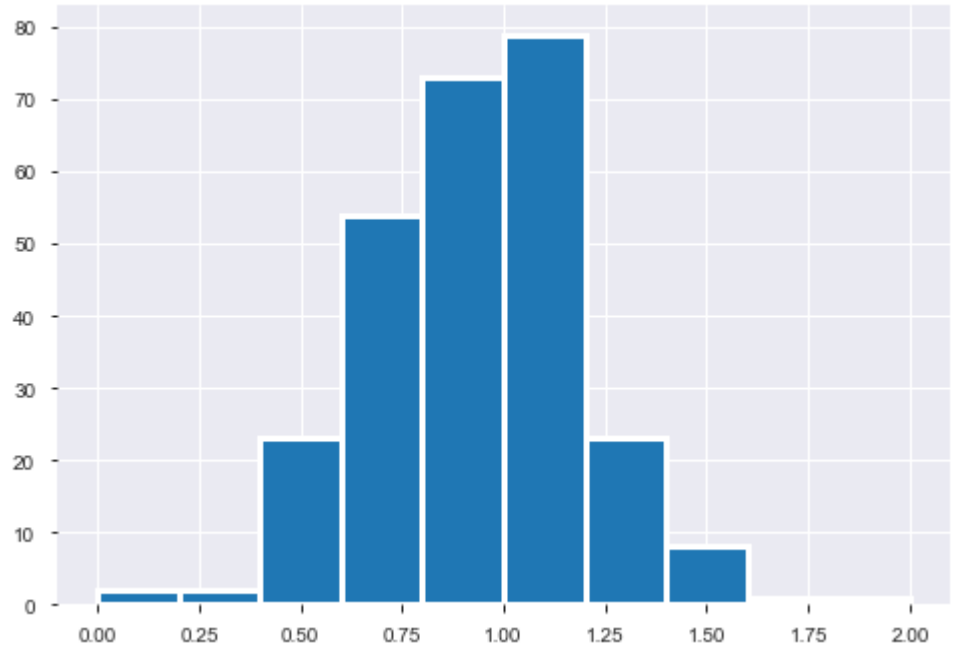
In [82]:
```
plt.hist(Produccion_df['Area Nacional (ha)'], edgecolor='white', linewidth=3)
# Histograma del Area Nacional
```

Out[82]:
```
(array([98., 44., 26., 27., 18., 11., 11., 13.,  7., 11.]),
 array([ 0.   ,  1.643,  3.286,  4.929,  6.572,  8.215,  9.858, 11.501,
        13.144, 14.787, 16.43 ]),
 <a list of 10 Patch objects>)
```

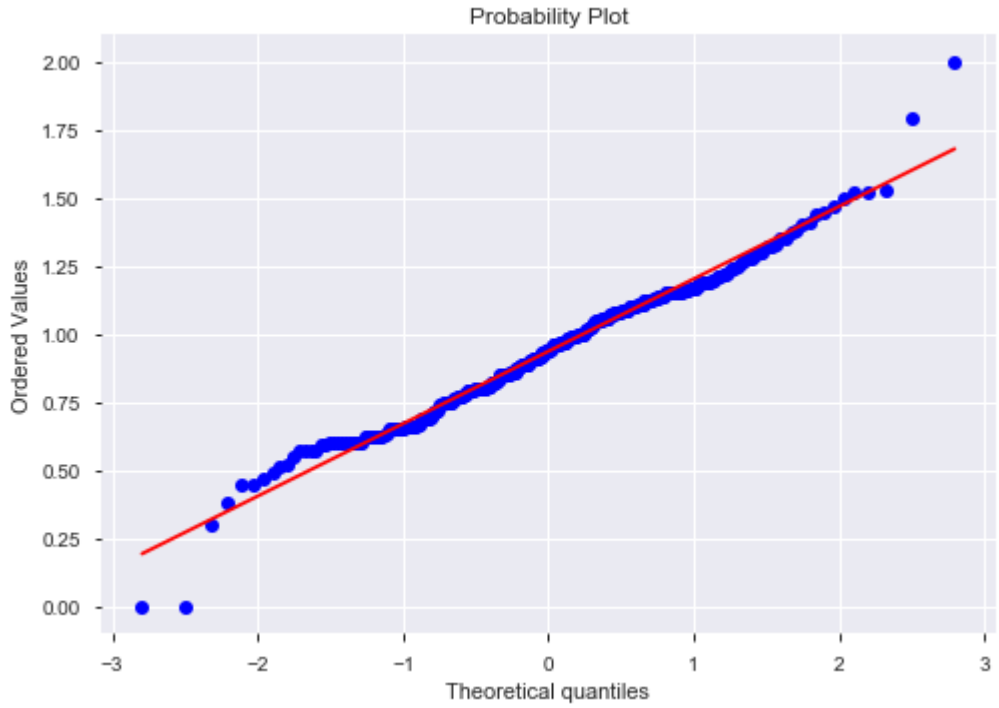


In [83]:
```
plt.hist(Produccion_df['Rendimiento (ha/ton)'], edgecolor='white', linewidth=3)
# Histograma del Rendimiento
```

Out[83]:
```
(array([ 2.,  2., 23., 54., 73., 79., 23.,  8.,  1.,  1.]),
 array([0. , 0.2, 0.4, 0.6, 0.8, 1. , 1.2, 1.4, 1.6, 1.8, 2. ]),
 <a list of 10 Patch objects>)
```



In [84]:
```
import pylab
import scipy.stats as stats #Librerias para construir estos tipos de graficos
stats.probplot(Produccion_df['Rendimiento (ha/ton)'], dist= 'norm', plot = pylab)
pylab.show()
```



In [85]:
```
from scipy.stats import shapiro
estadistico,p_value =shapiro(Produccion_df['Rendimiento (ha/ton)'])
print('Estadística=%.3f, El Valor de: p_value=%.3f' % (estadistico,p_value))
```

```
Estadística=0.983, El Valor de: p_value=0.003
```

In [86]:
```
produccion_correlacion_spearman = Produccion_df.corr(method='spearman')
produccion_correlacion_spearman
```

Out[86]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio	1.000000	0.180205	0.037725	0.023246
Rendimiento (ha/ton)	0.180205	1.000000	0.366952	0.264041
Produccion Nacional (ton)	0.037725	0.366952	1.000000	0.986380
Area Nacional (ha)	0.023246	0.264041	0.986380	1.000000

In [87]:
```
produccion_correlacion_pearson = Produccion_df.corr(method='pearson')
produccion_correlacion_pearson
```

Out[87]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio	1.000000	0.173474	0.007957	0.008715
Rendimiento (ha/ton)	0.173474	1.000000	0.385570	0.280677
Produccion Nacional (ton)	0.007957	0.385570	1.000000	0.978409
Area Nacional (ha)	0.008715	0.280677	0.978409	1.000000

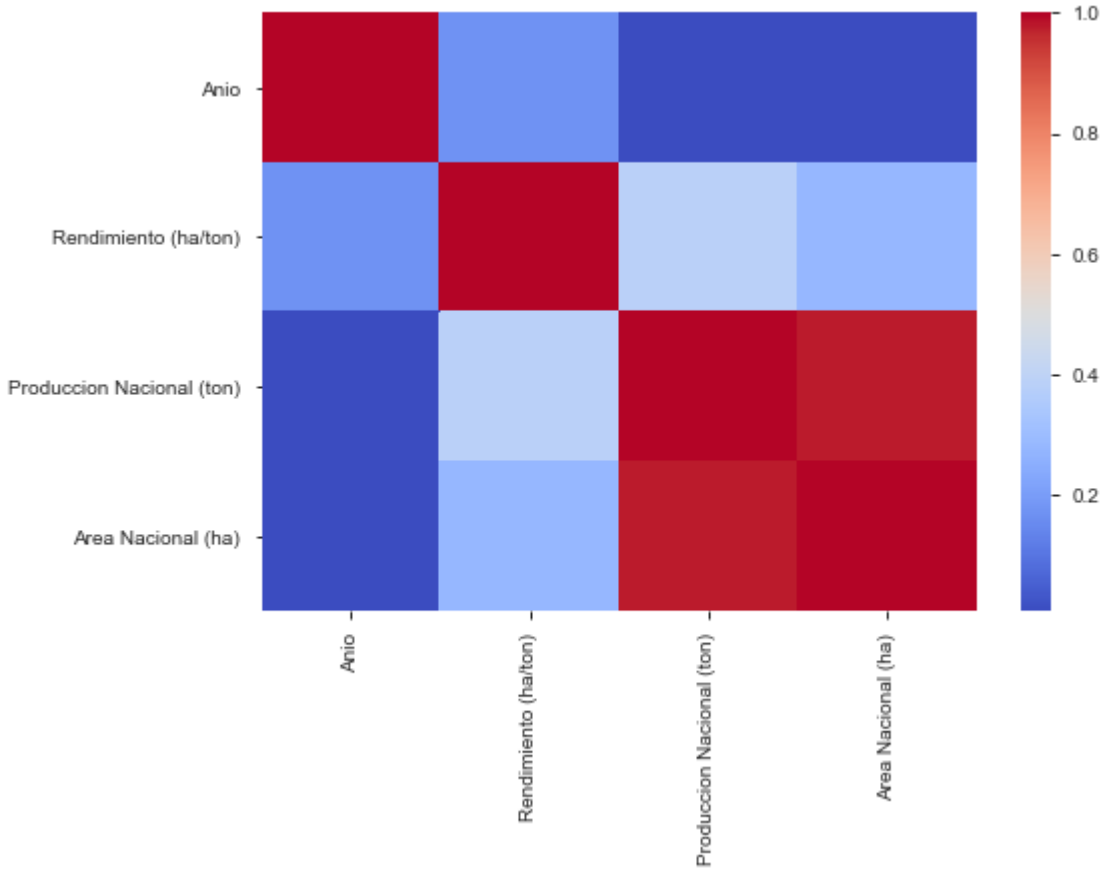
In [88]:
```
produccion_correlacion_kendall = Produccion_df.corr(method='kendall')
produccion_correlacion_kendall
```

Out[88]:

	Anio	Rendimiento (ha/ton)	Produccion Nacional (ton)	Area Nacional (ha)
Anio	1.000000	0.140836	0.026879	0.016567
Rendimiento (ha/ton)	0.140836	1.000000	0.265165	0.186979
Produccion Nacional (ton)	0.026879	0.265165	1.000000	0.909233
Area Nacional (ha)	0.016567	0.186979	0.909233	1.000000

In [89]:
```
sns.heatmap(produccion_correlacion_pearson,
xticklabels=produccion_correlacion_pearson.columns,
yticklabels=produccion_correlacion_pearson.columns,
cmap='coolwarm')
```

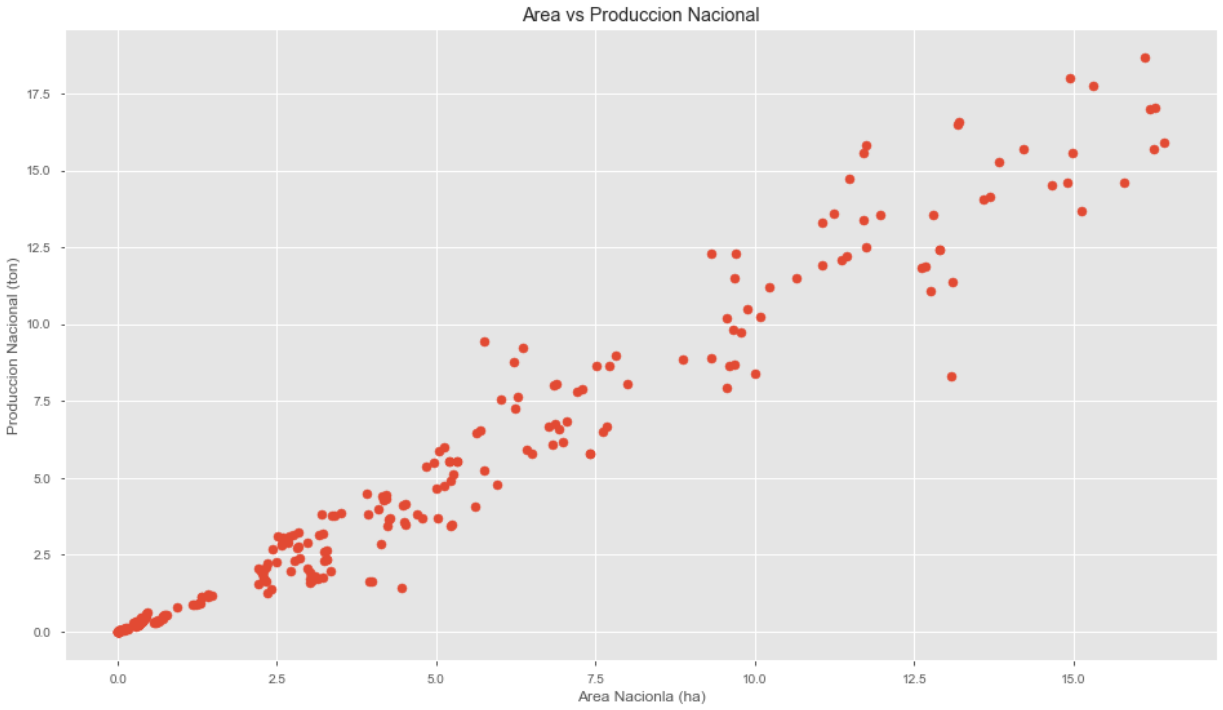
Out[89]: <matplotlib.axes._subplots.AxesSubplot at 0x122a5a08>



In [90]:
```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm
plt.rcParams['figure.figsize'] = (16, 9)
plt.style.use('ggplot')
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

In [91]:
```
plt.scatter(Produccion_df['Area Nacional (ha)'],Produccion_df['Produccion Naciona
plt.title('Area vs Produccion Nacional')
plt.xlabel('Area Nacionla (ha)')
plt.ylabel("Produccion Nacional (ton)")
```

Out[91]: Text(0, 0.5, 'Produccion Nacional (ton)')



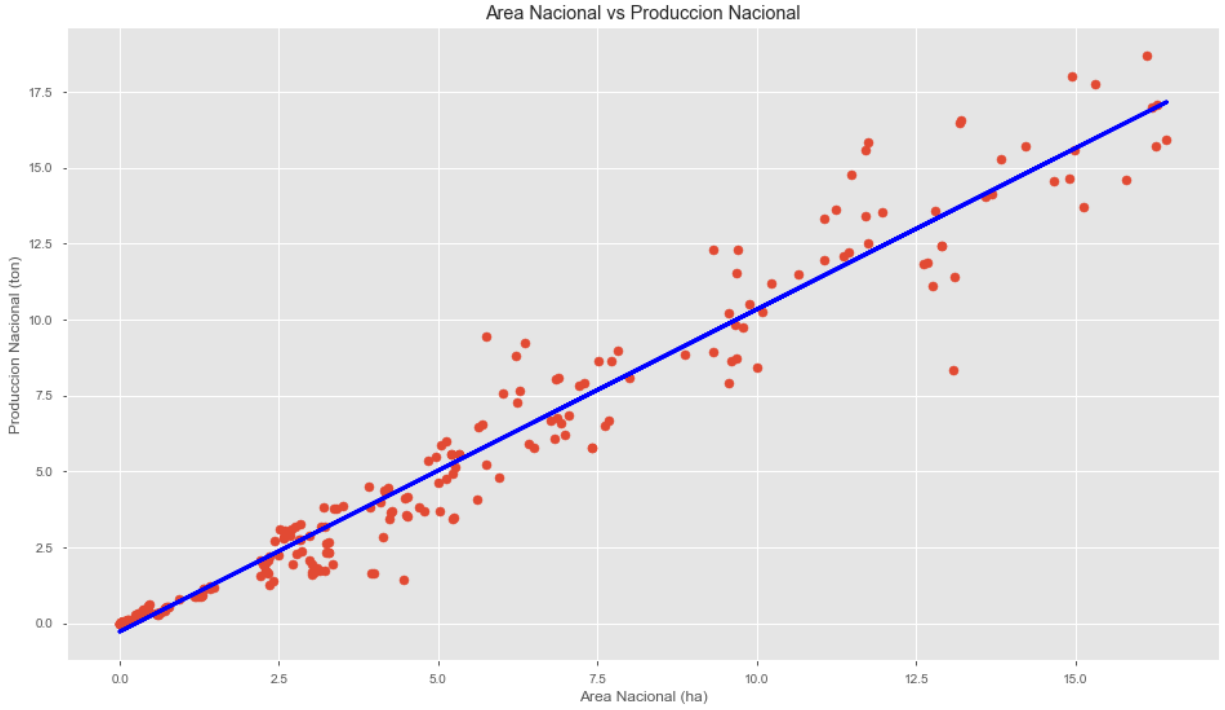
In [92]:
```
# Iniciamos el proceso para determinar el modelo de regresion lineal, de la anali
# Asignamos a nuestra variable de entrada X (En este caso corresponde al Area Nac
# Asignamos a la variable dependiente Y (En este caso corresponde a la Produccior
dataX =Produccion_df[["Area Nacional (ha)"]]
X_train = np.array(dataX)
y_train = Produccion_df['Produccion Nacional (ton)'].values
```

In [93]:
```
# Creamos la función objeto para determinar la Regresión Lineal Y= mX+bo
regr = linear_model.LinearRegression()
# Entrenamos nuestro modelo de regresion lineal, con la siguiente función
regr.fit(X_train, y_train)
# Hacemos las predicciones segun el modelo de regresion lineal
y_pred = regr.predict(X_train)

print('Valor de la tangente (m) o Coefficients:=======> ', regr.coef_)
# Ahora el valor de la constante bo, es decir el valor donde la recta corta el ej
print('Valor de la constante o Independent term: =====>', regr.intercept_)
# Se imprime el Error Cuadrado Medio
```

```
Valor de la tangente (m) o Coefficients:=======>  [1.06084584]
Valor de la constante o Independent term: =====> -0.2743751434833568
```

In [94]:
```
plt.scatter(Produccion_df['Area Nacional (ha)'],Produccion_df['Produccion Nacional
plt.title('Area Nacional vs Produccion Nacional')
plt.xlabel('Area Nacional (ha)')
plt.ylabel('Produccion Nacional (ton)')
# A continuación se grafica en colo azul, la funcion lineal obtenida a partir del
plt.plot(X_train[:,0], y_pred, color='Blue', linewidth=3)
plt.show()
```



In [95]:
```
# Ahora vamos a predecir utilizando la función obtenida, la producción nacional (
# Queremos predecir cuántos toneladas de producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[2]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %produ
```

```
Estimación de la Producción Nacional del Café en toneladas===>1.847
```

In [96]:
```
# Ahora vamos a predecir utilizando la función obtenida, la producción nacional (
# Queremos predecir cuántos toneladas de producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[2.5]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %produ
```

```
Estimación de la Producción Nacional del Café en toneladas===>2.378
```

In [97]:
```
# Ahora vamos a predecir utilizando la función obtenida, la producción nacional (
# Queremos predecir cuántos toneladas de producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[8]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %produ
```

```
Estimación de la Producción Nacional del Café en toneladas===>8.212
```

In [98]:
```
# Ahora vamos a predecir utilizando la función obtenida, la producción nacional (
# Queremos predecir cuántos toneladas de producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[11]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %produ
```

```
Estimación de la Producción Nacional del Café en toneladas===>11.395
```

In [99]:
```
# Ahora vamos a predecir utilizando la función obtenida, la producción nacional (
# Queremos predecir cuántos toneladas de producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[15]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %produ
```

```
Estimación de la Producción Nacional del Café en toneladas===>15.638
```

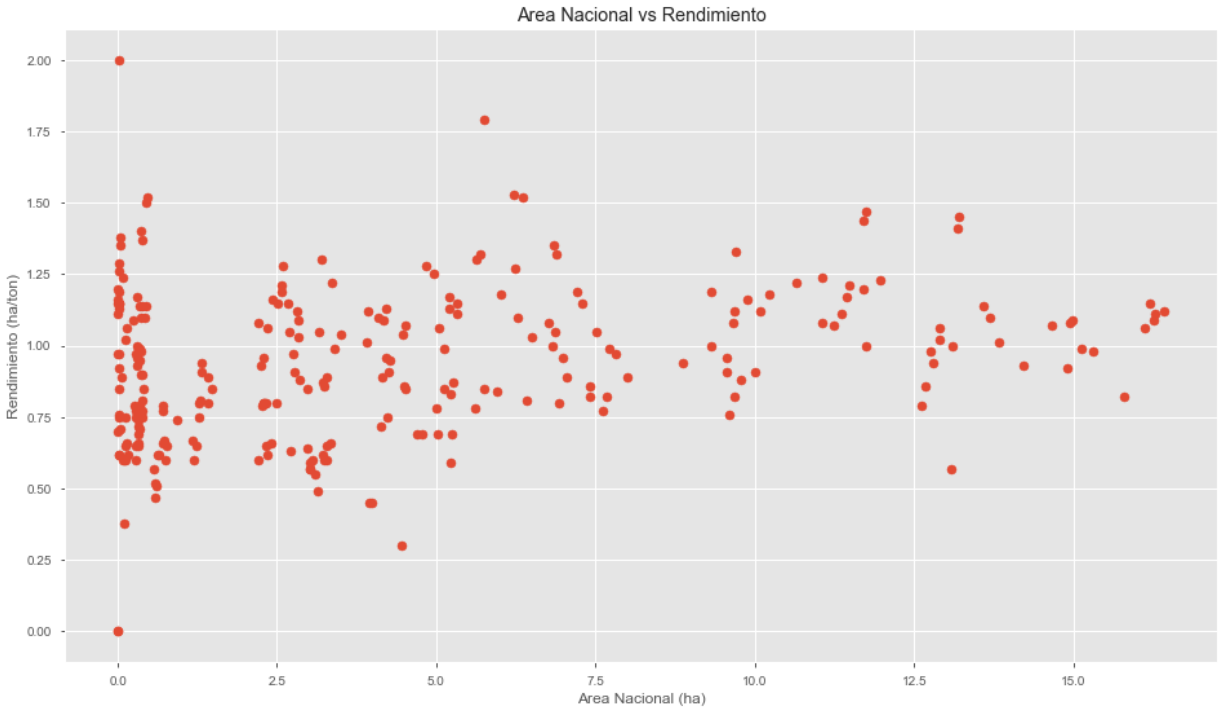
In [100]:
```
# Ahora vamos a predecir utilizando la función obtenida, la producción nacional (
# Queremos predecir cuántos toneladas de producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[35]])
print('Estimación de la Producción Nacional del Café en toneladas===>%.3f' %produ
```

```
Estimación de la Producción Nacional del Café en toneladas===>36.855
```

In [101]:
```
# Así,de esta manera, ya conociendo el proceso de analitica deterministica, podem
# Iniciamos el proceso para determinar el modelo de regresion lineal
# Asignamos a nuestra variable de entrada X (En este caso corresponde al Area Nac
# Asignamos a la variable dependiente Y (En este caso corresponde a Rendimiento)
dataX =Produccion_df[["Area Nacional (ha)"]]
X_train = np.array(dataX)
y_train = Produccion_df['Rendimiento (ha/ton)'].values
```

In [102]:
```
# Gráfico de dispersion del comportamiento del Area Nacional versus Produccion Na
plt.scatter(Produccion_df['Area Nacional (ha)'],Produccion_df['Rendimiento (ha/to
plt.title('Area Nacional vs Rendimiento')
plt.xlabel('Area Nacional (ha)')
plt.ylabel('Rendimiento (ha/ton)')
```

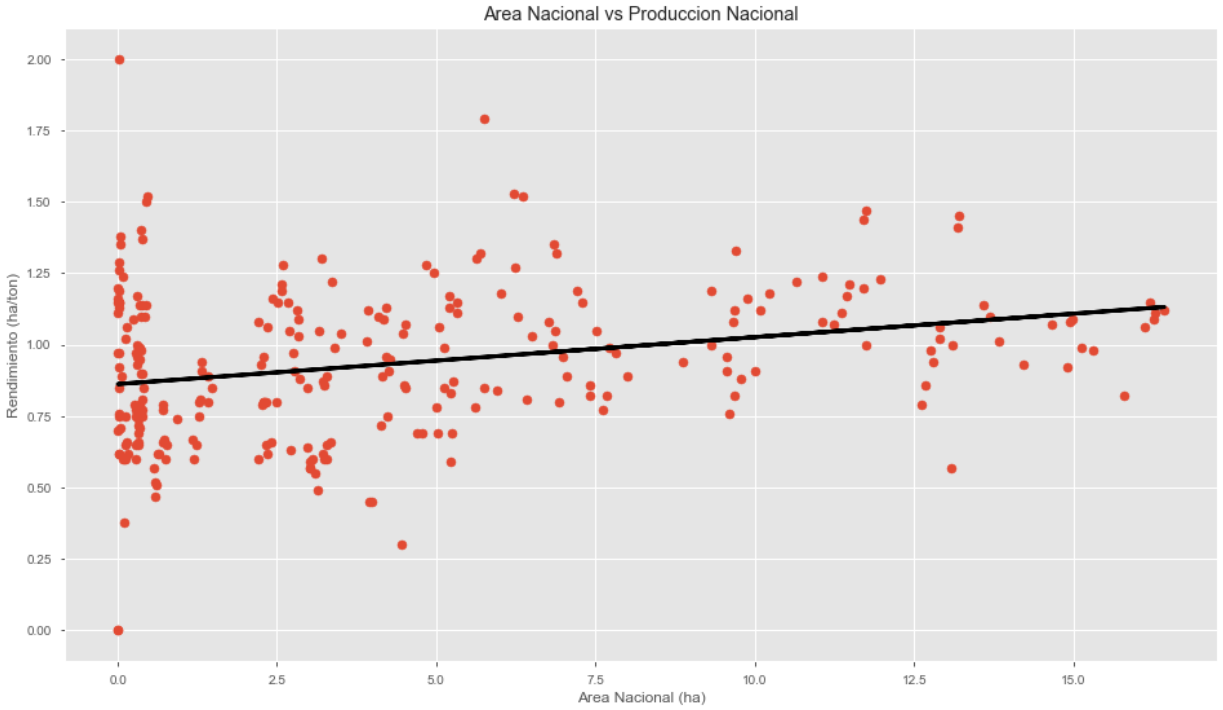
Out[102]:
```
Text(0, 0.5, 'Rendimiento (ha/ton)')
```



In [103]:
```
# Creamos la función objeto para determinar la Regresión Lineal Y= mX+bo
regr = linear_model.LinearRegression()
# Entrenamos nuestro modelo de regresion lineal, con la siguiente función
regr.fit(X_train, y_train)
# Hacemos las predicciones segun el modelo de regresion lineal
y_pred = regr.predict(X_train)
# Ahora imprimimos los resultados obtenidos
# Vemos el valor de la pendiente, osea la variable m, el coeficiente de la variab
print('Valor de la tangente (m) o Coefficients:=======> ', regr.coef_)
# Ahora el valor de la constante bo, es decir el valor donde la recta corta el ej
print('Valor de la constante o Independent term: =====>', regr.intercept_)
```

```
Valor de la tangente (m) o Coefficients:=======>  [0.01642121]
Valor de la constante o Independent term: =====> 0.8623491800082033
```

In [104]:
```
# Gráfico de dispersion del comportamiento del Area Nacional versus Produccion Na
plt.scatter(Produccion_df['Area Nacional (ha)'],Produccion_df['Rendimiento (ha/to
plt.title('Area Nacional vs Produccion Nacional')
plt.xlabel('Area Nacional (ha)')
plt.ylabel("Rendimiento (ha/ton)")
# A continuación se grafica en colo azul, la funcion lineal obtenida a partir del
plt.plot(X_train[:,0], y_pred, color='black', linewidth=3)
plt.show()
```



In [105]:
```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton):
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[2]])
print('Estimación del rendimiento del Café en (hectareas/toneladas)===>%.3f' %prc
```

```
Estimación del rendimiento del Café en (hectareas/toneladas)===>0.895
```

In [106]:
```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton):
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[6]])
print('Estimación del rendimiento del Café en (hectareas/toneladas)===>%.3f' %pro
```

```
Estimación del rendimiento del Café en (hectareas/toneladas)===>0.961
```

In [107]:
```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton):
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[11]])
print('Estimación del rendimiento del Café en (hectareas/toneladas)===>%.3f' %pro
```

```
Estimación del rendimiento del Café en (hectareas/toneladas)===>1.043
```

In [108]:
```
# Ahora vamos a predecir utilizando la función obtenida, eL RENDIMIENTO (ha/ton):
# Queremos predecir el rendimiento de la producción nacional de Café vamos a obte
# según nuestro modelo, hacemos:
produccion_obtenida = regr.predict([[26]])
print('Estimación del rendimiento del Café en (hectareas/toneladas)===>%.3f' %pro
```

```
Estimación del rendimiento del Café en (hectareas/toneladas)===>1.289
```

In [ ]:
```

```