PRACTICA2. Tipologia de Datos

Preparación y analisis de un dataset

Script de preparación y análisis del Dataset "Wine Review". El objetivo planteado es verificar si existen relaciones significativas entre entre el precio de un vino, su valoración y la descripción que del mismo realiza un sommelier experto. Para simplificar el análisis, se tomará como elemento de valoración de la descripción la longitud de la misma. De este modo el analisis se centrará exclusivamente en valores numericos.

La numeración dada a cada apartado se corresponde con la del documento explicativo.

2.6.1. Análisis de conjunto

El proceso de análisis de conjunto identificará el número total de registros (muestras) del dataset, así como los atributos, sus tipos y las estadisticas generales de los datos (promedios, varianza, etc.).

In [13]:
```
%matplotlib inline

import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from numpy import percentile
from matplotlib import pyplot
from scipy import stats
from scipy.stats import normaltest

# Cargamos los datos del fichero "winemag-data-130k-v2.csv" en un dataframe

path = "C:/Users/Usuario/CloudStation/02 Emprender/03 HolaParlem/10 Biblioteca/06 Master Data Science UOC/97 Entregas/
06 Tipologia de Datos/PRACTICA2/Dataset"
file_name="winemag-data-130k-v2.csv"
data = pd.read_csv(path + "/" + file_name)

def info(df):
    # Mostramos el número de muestras del fichero
    print ("Número de registros: " + str(len(df)) + "\n")

    # Mostramos el número de muestras del fichero
    print ("Número de atributos: " + str(len(df.columns)) + "\n")

    # Mostramos los atributos junto con el tipo asignado en la extracción
    print "Analisis del tipado de los datos:"
    print(df.dtypes)
    print ("\n")

    # Valores únicos de cada categoría/atributo
    print "Valores únicos por categoría:"
    print(data.nunique())
    print ("\n")

    # Analisis estadistico básico
    print "Analisis estadistico básico:"
    print (df.describe())
    print ("\n")
```

```
info(data)

# Visualizamos algunos datos (filas i a j)
print(data[1:10])
print ("\n")

# Nombramos los atributos que no están identificados (ID)
data.rename(columns={'Unnamed: 0':'id'}, inplace=True)
print (data.columns)
print ("\n")
```

```
Número de registros: 129971

Número de atributos: 14

Analisis del tipado de los datos:
Unnamed: 0                int64
country                  object
description              object
designation              object
points                    int64
price                   float64
province                 object
region_1                 object
region_2                 object
taster_name              object
taster_twitter_handle    object
title                    object
variety                  object
winery                   object
dtype: object


Valores únicos por categoría:
Unnamed: 0              129971
country                     43
description             119955
designation              37979
points                      21
price                      390
province                   425
region_1                  1229
region_2                    17
taster_name                 19
taster_twitter_handle       15
title                   118840
variety                    707
winery                   16757
dtype: int64


Analisis estadistico básico:
```

```
          Unnamed: 0         points          price
count  129971.000000  129971.000000  120975.000000
mean    64985.000000      88.447138      35.363389
std     37519.540256       3.039730      41.022218
min         0.000000      80.000000       4.000000
25%     32492.500000      86.000000      17.000000
50%     64985.000000      88.000000      25.000000
75%     97477.500000      91.000000      42.000000
max    129970.000000     100.000000    3300.000000


   Unnamed: 0   country                                        description  \
1           1  Portugal  This is ripe and fruity, a wine that is smooth...
2           2        US  Tart and snappy, the flavors of lime flesh and...
3           3        US  Pineapple rind, lemon pith and orange blossom ...
4           4        US  Much like the regular bottling from 2012, this...
5           5     Spain  Blackberry and raspberry aromas show a typical...
6           6     Italy  Here's a bright, informal red that opens with ...
7           7    France  This dry and restrained wine offers spice in p...
8           8   Germany  Savory dried thyme notes accent sunnier flavor...
9           9    France  This has great depth of flavor with its fresh ...

                         designation  points  price           province  \
1                            Avidagos      87   15.0              Douro
2                                 NaN      87   14.0             Oregon
3                 Reserve Late Harvest      87   13.0           Michigan
4  Vintner's Reserve Wild Child Block      87   65.0             Oregon
5                        Ars In Vitro      87   15.0     Northern Spain
6                             Belsito      87   16.0  Sicily & Sardinia
7                                 NaN      87   24.0             Alsace
8                               Shine      87   12.0        Rheinhessen
9                        Les Natures      87   27.0             Alsace

              region_1           region_2         taster_name  \
1                  NaN                NaN          Roger Voss
2    Willamette Valley  Willamette Valley        Paul Gregutt
3  Lake Michigan Shore                NaN  Alexander Peartree
4    Willamette Valley  Willamette Valley        Paul Gregutt
5              Navarra                NaN   Michael Schachner
6              Vittoria               NaN       Kerin O’Keefe
7                Alsace               NaN          Roger Voss
8                   NaN               NaN  Anna Lee C. Iijima
```

```
9              Alsace               NaN         Roger Voss

  taster_twitter_handle                                          title  \
1            @vossroger      Quinta dos Avidagos 2011 Avidagos Red (Douro)
2           @paulgwine       Rainstorm 2013 Pinot Gris (Willamette Valley)
3                   NaN  St. Julian 2013 Reserve Late Harvest Riesling ...
4           @paulgwine   Sweet Cheeks 2012 Vintner's Reserve Wild Child...
5           @wineschach  Tandem 2011 Ars In Vitro Tempranillo-Merlot (N...
6          @kerinokeefe   Terre di Giurfo 2013 Belsito Frappato (Vittoria)
7            @vossroger             Trimbach 2012 Gewurztraminer (Alsace)
8                   NaN  Heinz Eifel 2013 Shine Gewürztraminer (Rheinhe...
9            @vossroger  Jean-Baptiste Adam 2012 Les Natures Pinot Gris...

              variety               winery
1      Portuguese Red  Quinta dos Avidagos
2          Pinot Gris            Rainstorm
3            Riesling           St. Julian
4          Pinot Noir         Sweet Cheeks
5  Tempranillo-Merlot               Tandem
6            Frappato      Terre di Giurfo
7      Gewürztraminer             Trimbach
8      Gewürztraminer          Heinz Eifel
9          Pinot Gris   Jean-Baptiste Adam


Index([u'id', u'country', u'description', u'designation', u'points', u'price',
       u'province', u'region_1', u'region_2', u'taster_name',
       u'taster_twitter_handle', u'title', u'variety', u'winery'],
      dtype='object')
```

Conclusiones:

1.- El número total de registros es de: 129.971 2.- El número de vinos diferentes es muy alto (prácticamente hay un registro por vino) 3.- 707 variedades de uva 4.- Vinos de hasta 43 paises diferentes 5.- Los tipos de las variables numericas asignadas por defecto son coherentes a su función. No se realizarán cambios en ellas. 6.- Se nombra el atributo (de origen vacio), cuya función es identificar cada registro como 'id'. 7.- Todas las puntuaciones de los vinos identificados son superiores a 80, que es consistente con la información proporcionada en la página origen de los datos. 8.- El precio se encuentra concentrado en una horquilla de hasta 50 €, observandose la existencia de valores muy por debajo y muy por encima de la media. En consecuencia el análisis de valores extremos debe ser considerado particularmente relevante en este caso.

2.6.4. Limpieza

2.6.4.1.- Selección de atributos

A priori el 'taster_name', el 'taster_twitter_handle', 'region_1', 'region_2' podrian eliminarse del dataset para su simplificación, ya que son variables que no contemplamos dentro de este análisis y de esta forma aligeramos el tratamiento de datos. Hay otras variables que tampoco forman parte del análisis pero preferimos no eliminarlas puesto que consideramos que pueden ser utiles de cara a completar posibles valores faltantes. Por otra parte incorporamos la longitud del texto de la descripcion como variable adicional.

In [14]:
```
# Separamos los registros que no son de interés del dataset, eliminandolas

del data['taster_name']
del data['taster_twitter_handle']
del data['region_1']
del data['region_2']
data['descrip_length'] =data.description.str.len()
data.head(5)
```

Out[14]:

	id	country	description	designation	points	price	province	title	variety	winery	descrip_length
0	0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Nicosia 2013 Vulkà Bianco (Etna)	White Blend	Nicosia	172
1	1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	Quinta dos Avidagos 2011 Avidagos Red (Douro)	Portuguese Red	Quinta dos Avidagos	227
2	2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Rainstorm 2013 Pinot Gris (Willamette Valley)	Pinot Gris	Rainstorm	186
3	3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	St. Julian 2013 Reserve Late Harvest Riesling ...	Riesling	St. Julian	199
4	4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Sweet Cheeks 2012 Vintner's Reserve Wild Child...	Pinot Noir	Sweet Cheeks	249

2.6.4.2.- Duplicados

Buscamos si existen muestras duplicadas.

In [15]:
```
# Creamos una fila que identifique si hay duplicados
data['esta_duplicado']= data.duplicated()
print("El número de muestras duplicadas es de %d.") %len(data[data['esta_duplicado']==True])
# Eliminamos la fila creada
del data['esta_duplicado']
```

```
El número de muestras duplicadas es de 0.
```

2.6.4.3.- Análisis de registros vacíos o nulos

En primer lugar identificamos el número de registros vacios para cada atributo:

In [16]:
```
# Contamos los registros vacios y calculamos el porcentaje que representan
def vacios(df):
    muestras_vacias = df.isnull().sum()
    # identificamos su peso en cada caso:
    total_registros = np.product(df.shape)
    total_vacias = muestras_vacias.sum()
    return ((float(total_vacias)/float(total_registros)) * 100)


print ("El numero de muestras vacias representa el %f porciento del total.") %vacios(data)


# verificamos que los campos vacios encontrados tanto en 'country' como en 'province' corresponden a las mismas muestr
as,
#al observar que ambos campos tienen el mismo numero de registros vacios.
if (len(data.loc[data.title.isin(data[data.country.isnull() & data.province.isnull()].title)])==len(data[data.country.
isnull()])):
    print('Las muestras de country vacias tambien tienen province vacio')
else:
    print('Las muestras de country vacias NO coinciden necesariamente con province vacio')

# verificamos si podemos completar la información de variedad faltante (dado que solo hay un registro)
# a partir del texto de descripcion o del titulo del vino
pd.set_option('display.max_colwidth', -1)
print(data[data.variety.isnull()].title)
print(data[data.variety.isnull()].description)
pd.set_option('display.max_colwidth', 50)
```

```
El numero de muestras vacias representa el 3.258629 porciento del total.
Las muestras de country vacias tambien tienen province vacio
86909    Carmen 1999  (Maipo Valley)
Name: title, dtype: object
86909    A chalky, dusty mouthfeel nicely balances this Petite Syrah's bright, full blackberry and blueberry fruit. W
heat-flour and black-pepper notes add interest to the bouquet; the wine finishes with herb and an acorny nuttiness. A
good first Chilean wine for those more comfortable with the Californian style. It's got tannins to lose, but it's ver
y good.
Name: description, dtype: object
```

Conclusiones:

1.- Solo el 3,6% de los registros estan vacios 2.- El mayor número de muestras vacias corresponde a la característica 'designation' (viñedo) 3.- Todas las muestras tienen valoracion (puntos y descripción) 4.- No se dispone de precio para un total de 8996 registros 5.- Aquellos campos que no disponen de información del pais, tampoco disponen de información de provincia y el vino de dichos campos, identificado por su titulo es único y en consecuencia no puede ser completada esta información a partir de otros valores. 6.- Solo un registro no tiene información de variedad (tipo de uva) y no observamos que podamos completarlo con la información de otros registros.

Decisiones:

1.- Eliminamos todos aquellos registros con algun campo de interés nulo. Dado que 'designation' no va a formar parte de dicho estudio, para evitar reducir en exceso el dataset, lo completamos como 'Unknow' y así lo mantendremos posteriormente al eliminar el resto que son vacios.

In [17]:
```
# Etiquetamos como desconocidos los campos vacios del atributo designation para mantenerlos
data['designation'] = data.designation.replace(np.NaN, 'Unknown')
# Eliminamos los registros vacios distinguiendo los casos en que los atributos son numericos o cadenas de texto.
atributtes=list(data)
for column_name in atributtes:
    if ((column_name != 'id') | (column_name != 'points') | (column_name != 'price')):
        data = data[pd.notnull(data[column_name])]
    else:
        data.dropna(axis=0, subset=[column_name])

# Comprobamos que hemos eliminado los registros vacios
info(data)
print ("El numero de muestras vacias representa el %f por ciento del total.") %vacios(data)
```

Número de registros: 120915

Número de atributos: 11

Analisis del tipado de los datos:

```
id                 int64
country           object
description       object
designation       object
points             int64
price            float64
province          object
title             object
variety           object
winery            object
descrip_length     int64
dtype: object
```

Valores únicos por categoría:

```
id               120915
country             42
description      111511
designation       35754
points               21
price               390
province            422
title            110582
variety             691
winery            15843
descrip_length      566
dtype: int64
```

Analisis estadistico básico:

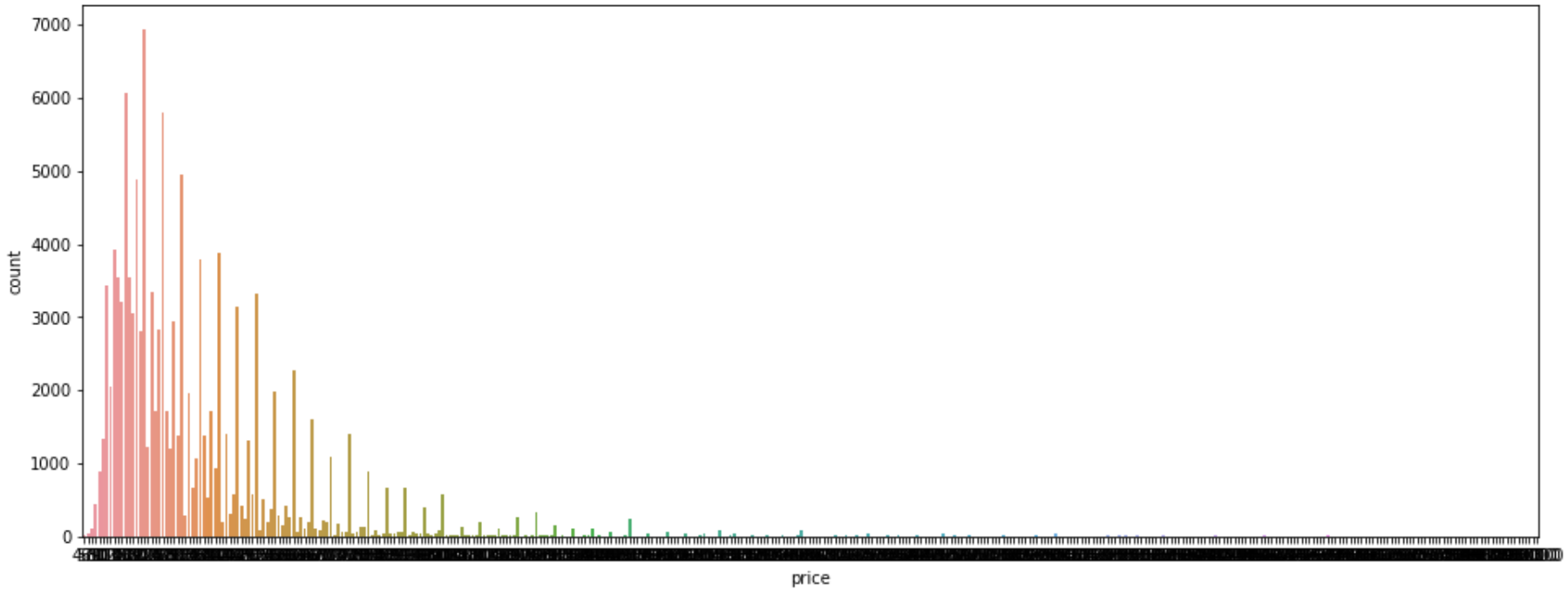
	id	points	price	descrip_length
count	120915.000000	120915.000000	120915.000000	120915.000000
mean	65043.605541	88.421726	35.368796	244.528826
std	37511.733783	3.044954	41.031188	66.792224
min	1.000000	80.000000	4.000000	20.000000
25%	32571.500000	86.000000	17.000000	199.000000

```
50%     65141.000000      88.000000      25.000000      239.000000
75%     97501.500000      91.000000      42.000000      284.000000
max    129970.000000     100.000000    3300.000000      829.000000


El numero de muestras vacias representa el 0.000000 por ciento del total.
```

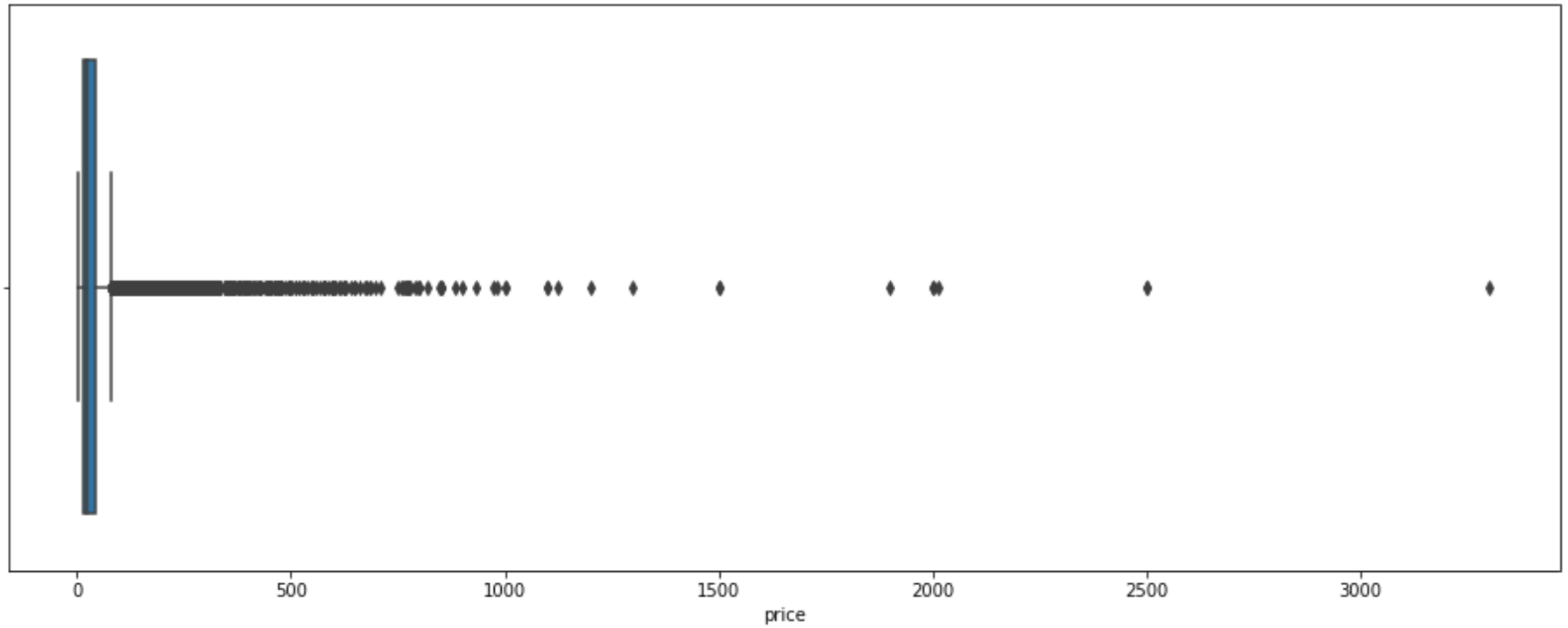
2.6.4.4.- Outliers

Procedemos a la identificación de valores extremos para los registros numericos de precio, valoracion y longitud del texto de descripcion. Lo hacemos mediante analisis visual y aplicando el test de D'Agostino's (se ha considerado este test puesto que el numero de registros disponibles es elevado)

In [18]:
```
# Outliers: analisis del registro de precio

#comprobamos visualmente la normalidad
plt.figure(figsize=(16,6))
plt.subplot(1,1,1)
g = sns.countplot(x='price', data=data)
plt.show()

#comprobamos la normalidad mediante test (D'Agostino's K^2 Test)
stat, p = normaltest(data['price'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpretamos el resultado
alpha = 0.05
if p > alpha:
    print('Las muestras parecen seguir una distribución normal')
else:
    print('Las muestras no siguen una distribución normal.')


#identificamos visualmente los outliers
plt.figure(figsize=(16,6))
plt.subplot(1,1,1)
sns.boxplot(x=data['price'])
plt.show()


# identificar los outliers considerando que la muestra no es normal. (Metodo de intercuartiles)
q25, q75 = percentile(data['price'], 25), percentile(data['price'], 75)
iqr = q75 - q25
print('Percentiles: 25th=%.3f, 75th=%.3f, IQR=%.3f' % (q25, q75, iqr))
# calculamos los puntos de corte de los outliers
cut_off = iqr * 1.5
lower_price, upper_price = q25 - cut_off, q75 + cut_off
# identificamos los outliers
outliers = [x for x in data['price'] if x < lower_price or x > upper_price]
print('Outliers: %d' % len(outliers))
```

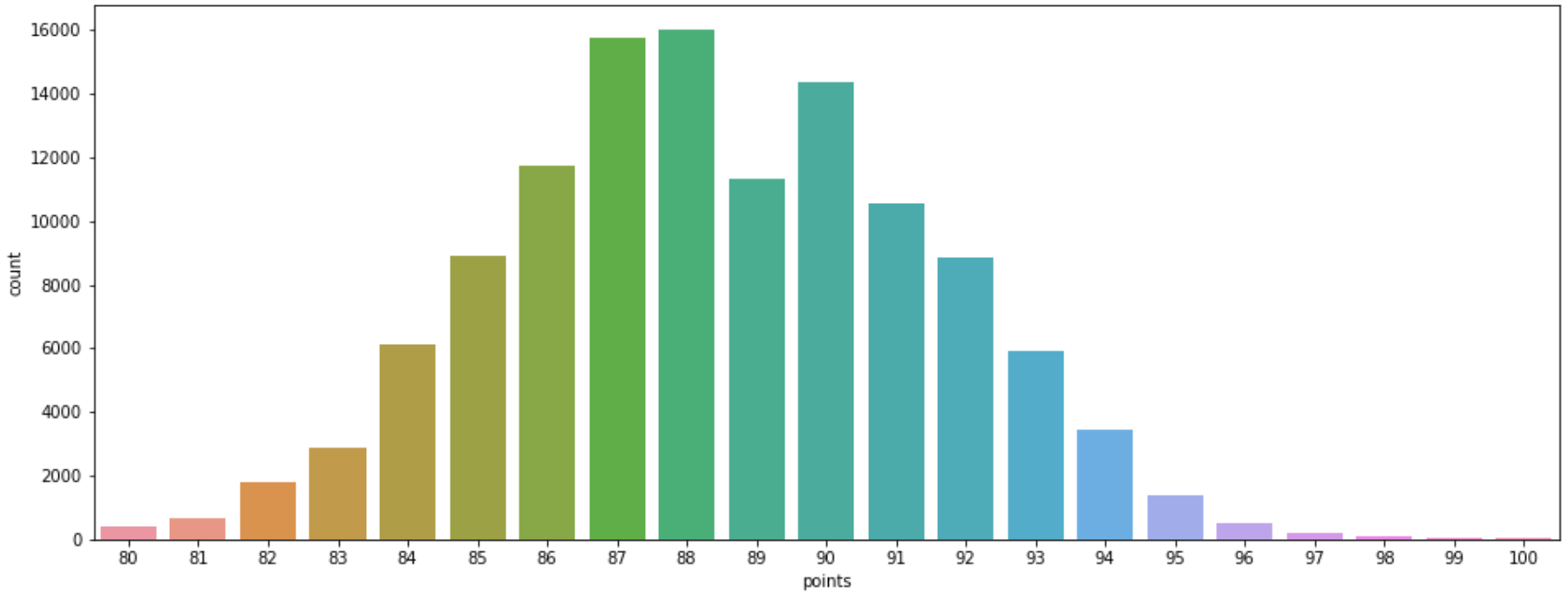


Statistics=255566.952, p=0.000
Las muestras no siguen una distribución normal.

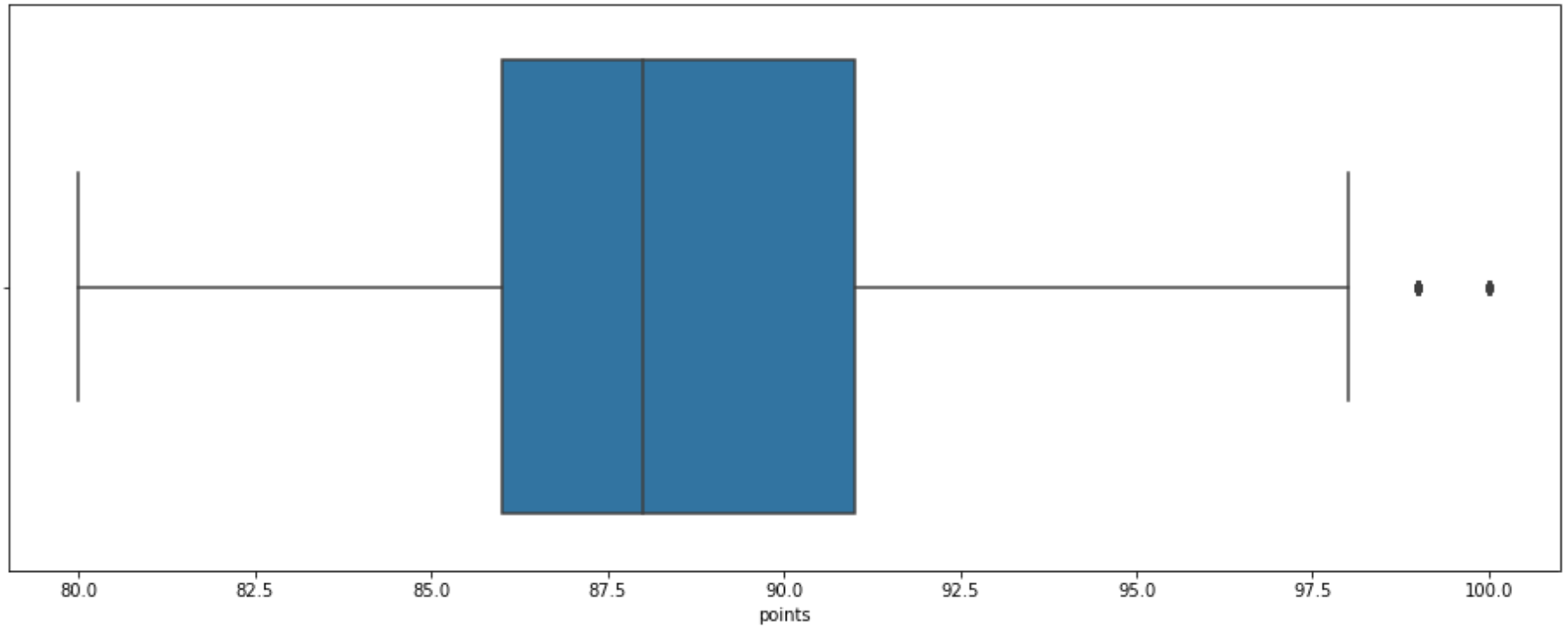


Percentiles: 25th=17.000, 75th=42.000, IQR=25.000
Outliers: 7241

In [19]:
```
# Outliers: analisis del registro de puntos

#comprobamos visualmente la normalidad
plt.figure(figsize=(16,6))
plt.subplot(1,1,1)
g = sns.countplot(x='points', data=data)
plt.show()

#comprobamos la normalidad mediante test (D'Agostino's K^2 Test)
stat, p = normaltest(data['points'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Las muestras parecen seguir una distribución normal')
else:
    print('Las muestras no siguen una distribución normal.')


#identificamos visualmente los outliers
plt.figure(figsize=(16,6))
plt.subplot(1,1,1)
sns.boxplot(x=data['points'])
plt.show()


# identificar los outliers considerando que la muestra no es normal. (Metodo de intercuartiles)
q25, q75 = percentile(data['points'], 25), percentile(data['points'], 75)
iqr = q75 - q25
print('Percentiles: 25th=%.3f, 75th=%.3f, IQR=%.3f' % (q25, q75, iqr))
# calculamos los puntos de corte de los outliers
cut_off = iqr * 1.5
lower_points, upper_points = q25 - cut_off, q75 + cut_off
# identificamos los outliers
outliers = [x for x in data['points'] if x < lower_points or x > upper_points]
print('Outliers: %d' % len(outliers))
```

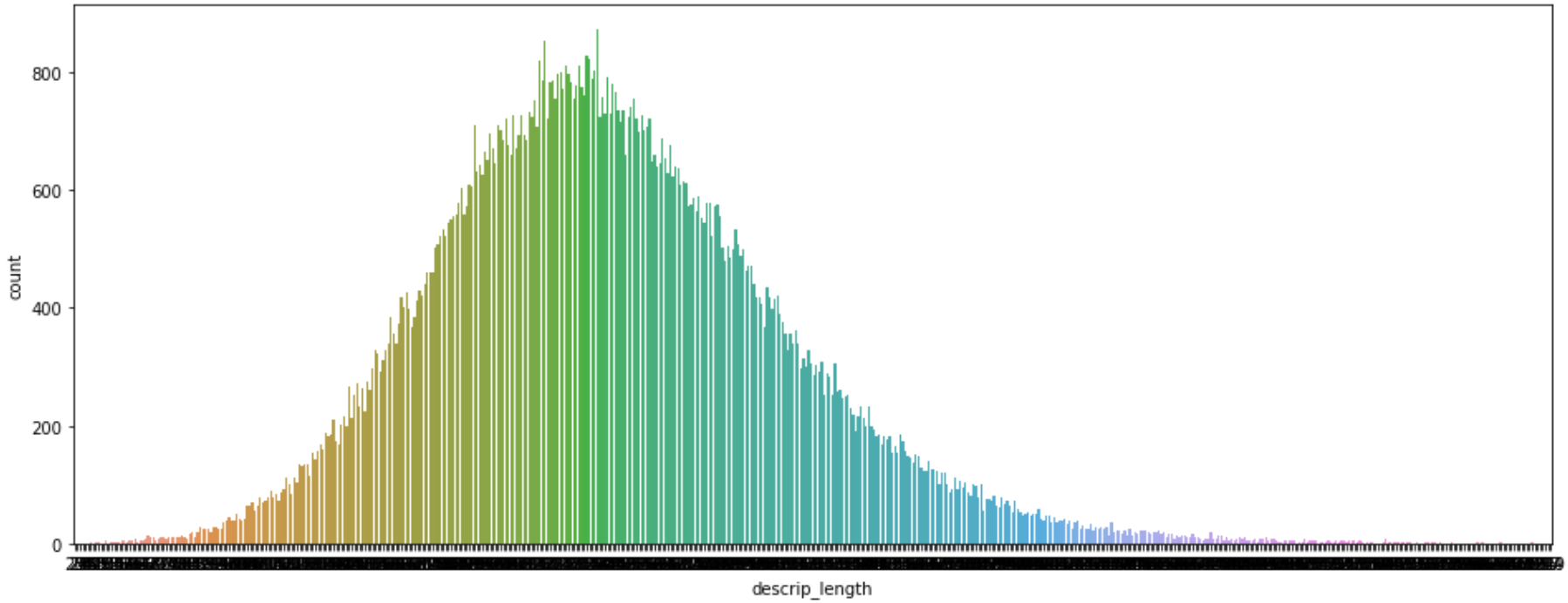


Statistics=625.634, p=0.000
Las muestras no siguen una distribución normal.

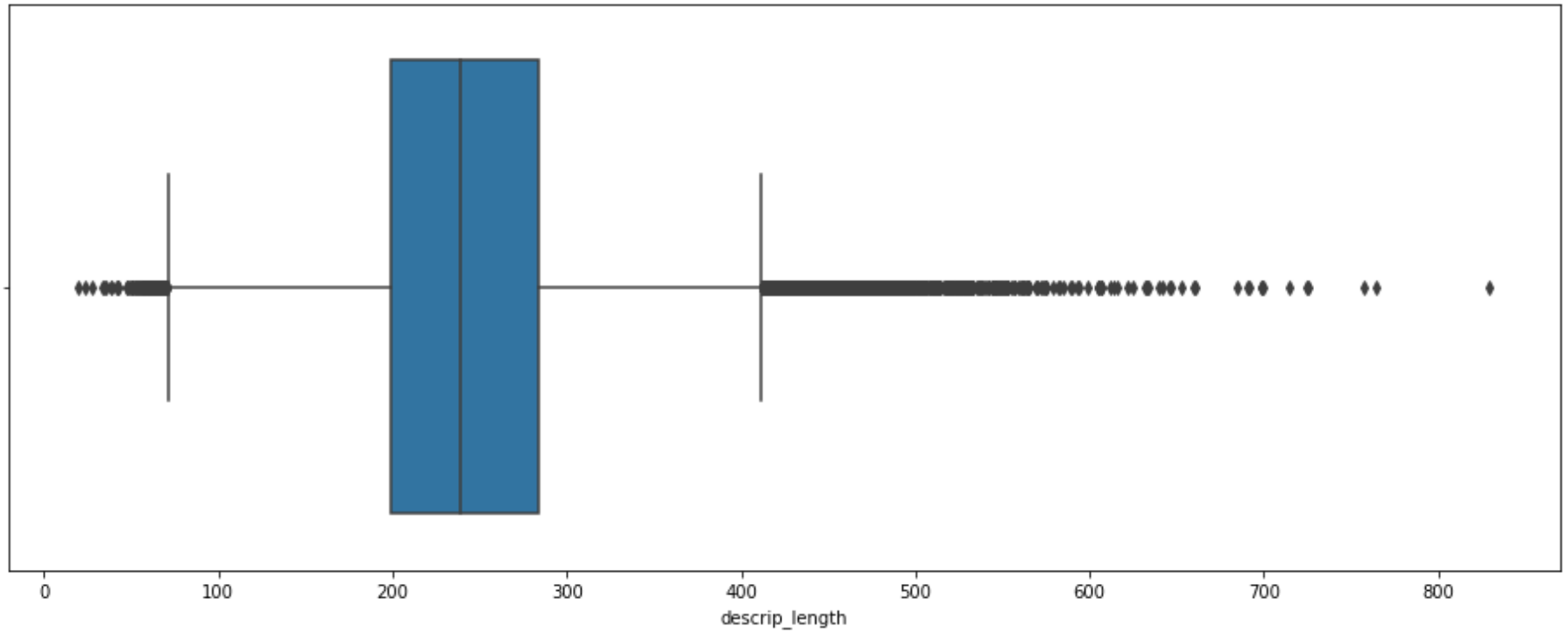


Percentiles: 25th=86.000, 75th=91.000, IQR=5.000
Outliers: 47

In [20]:
```
# Outliers: analisis del registro de longitud de la descripcion

#comprobamos visualmente la normalidad
plt.figure(figsize=(16,6))
plt.subplot(1,1,1)
g = sns.countplot(x='descrip_length', data=data)
plt.show()

#comprobamos la normalidad mediante test (D'Agostino's K^2 Test)
stat, p = normaltest(data['descrip_length'])
print('Statistics=%.3f, p=%.3f' % (stat, p))
# interpret
alpha = 0.05
if p > alpha:
    print('Las muestras parecen seguir una distribución normal')
else:
    print('Las muestras no siguen una distribución normal.')


#identificamos visualmente los outliers
plt.figure(figsize=(16,6))
plt.subplot(1,1,1)
sns.boxplot(x=data['descrip_length'])
plt.show()


# identificar los outliers considerando que la muestra no es normal. (Metodo de intercuartiles)
q25, q75 = percentile(data['descrip_length'], 25), percentile(data['descrip_length'], 75)
iqr = q75 - q25
print('Percentiles: 25th=%.3f, 75th=%.3f, IQR=%.3f' % (q25, q75, iqr))
# calculamos los puntos de corte de los outliers
cut_off = iqr * 1.5
lower_desc, upper_desc = q25 - cut_off, q75 + cut_off
# identificamos los outliers
outliers = [x for x in data['descrip_length'] if x < lower_desc or x > upper_desc]
print('Outliers: %d' % len(outliers))
```



Statistics=8771.603, p=0.000
Las muestras no siguen una distribución normal.




```
Percentiles: 25th=199.000, 75th=284.000, IQR=85.000
Outliers: 1961
```

Conclusiones

1.- Aunque visualmente tanto los atributos points como description_lenght parecen seguir una distribución normal, los test nos indican que no es así. 2.- Por ello hemos utilizado el método de intercuartiles para detectar los outliers. 3.- El número de outliers es muy elevado en el caso del precio (como ya se adelantaba) y también en el caso de las longitudes de la descripción.

Decisiones

De cara a decidir que hacer con dichos valores extremos, consideramos que la existencia de extremos el precio es admisible, considerando que en el mercado existen vinos con precios prohibitivos, sin embargo creemos que tanto la puntuación como la longitud de la descripción que a priori son valores subjetivos pueden conducir, de incorporarlos a los datos, a errores en el análisis.

In [21]:
```
# eliminamos los outliers de precio y longitud de descripcion obtenidos
data=data[((data['price']>= lower_price) & (data['price']<= upper_price))]
data=data[((data['descrip_length']>= lower_desc) & (data['descrip_length']<= upper_desc))]
info(data)
```

Número de registros: 112184

Número de atributos: 11

Analisis del tipado de los datos:

id	int64
country	object
description	object
designation	object
points	int64
price	float64
province	object
title	object
variety	object
winery	object
descrip_length	int64

dtype: object

Valores únicos por categoría:

id	112184
country	42
description	103212
designation	32987
points	20
price	76
province	420
title	102362
variety	684
winery	15476
descrip_length	340

dtype: int64

Analisis estadistico básico:

	id	points	price	descrip_length
count	112184.000000	112184.000000	112184.000000	112184.000000
mean	64978.984427	88.140261	28.958506	238.846734
std	37494.805895	2.870649	16.092037	60.565256
min	1.000000	80.000000	4.000000	72.000000
25%	32470.750000	86.000000	16.000000	196.000000

```
50%     65106.500000      88.000000      25.000000      236.000000
75%     97373.250000      90.000000      39.000000      279.000000
max    129970.000000      99.000000      79.000000      411.000000
```

2.6.4.5.- Salvamos el fichero preparado

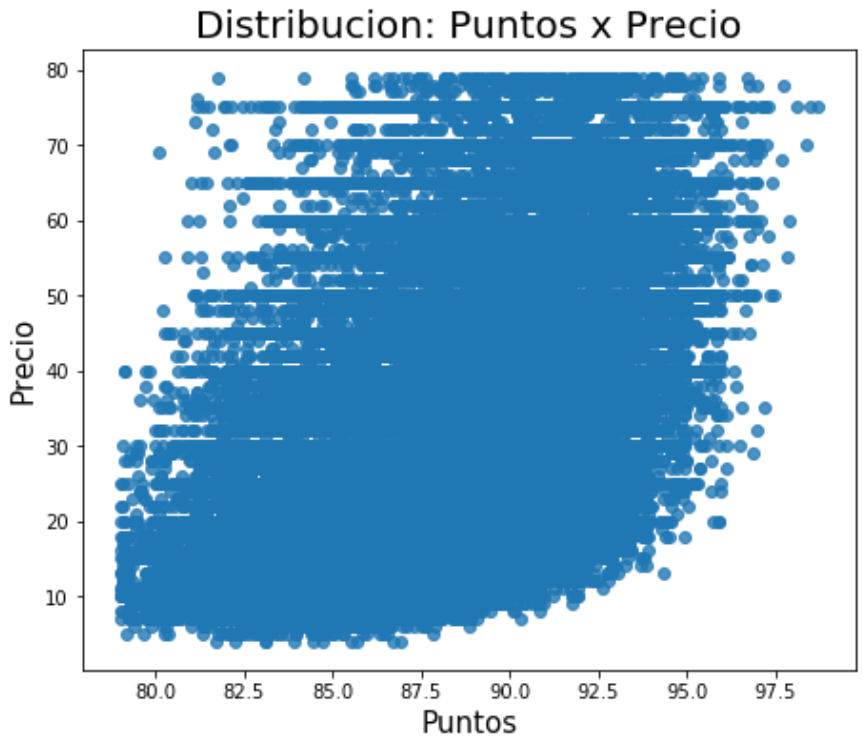
Procedemos a salvar el fichero con los datos preparados. El conjunto de registros final es de: 112184, con 11 atributos.

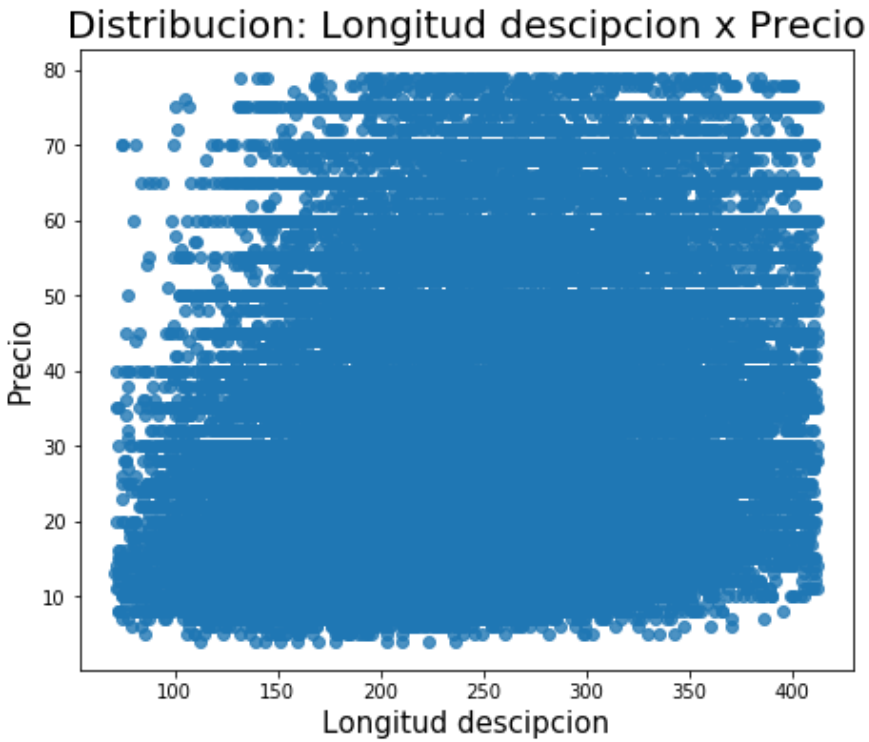
In [22]:
```
# Guardar un csv
data.to_csv("data_prepared.csv")
```

2.7.- Analisis

Para proceder al análisis de la relacion existente entre el precio, la valoración y la longitud de la descripción realizaremos un análisis de correlación. Ya hemos verificado previamente que ninguna de las variables presenta un comportamiento normal. Es por ello que realizaremos el cálculo del coeficiente de correlación de las diferentes variables con respecto al precio utilizando el coeficiente de Spearman.

In [23]:
```
#carga de librerias estadísticas (necesarias para el calculo del coeficiente de correlacion)
from scipy.stats import spearmanr

# cargamos el fichero con los datos preparados
path = "C:/Users/Usuario/00 Practicas UOC"
file_name="data_prepared.csv"
data = pd.read_csv(path + "/" + file_name)

#Analisis gráfico previo
plt.figure(figsize=(16,6))
plt.subplot(1,2,1)
g = sns.regplot(x='points', y='price', data=data, x_jitter=True, fit_reg=False)
g.set_title("Distribucion: Puntos x Precio", fontsize=20)
g.set_xlabel("Puntos", fontsize= 15)
g.set_ylabel("Precio", fontsize= 15)
plt.subplot(1,2,2)
g = sns.regplot(x='descrip_length', y='price', data=data, x_jitter=True, fit_reg=False)
g.set_title("Distribucion: Longitud descipcion x Precio", fontsize=20)
g.set_xlabel("Longitud descipcion", fontsize= 15)
g.set_ylabel("Precio", fontsize= 15)
plt.show()

#Calculo del coeficiente de cortrelación de spearman
corr, p_value = spearmanr(data['price'], data['points'])
print corr

corr, p_value = spearmanr(data['price'], data['descrip_length'])
print corr
```






```
0.56110164333
0.331556718665
```

Conclusiones

Considerando que en el rango [-1,1] ambos valores son positivos, observamos que en ambos casos hay influencia entre las variables, si bien el atributo de valoracion es mas infuyente en el precio que la longitd de la descripción.