

Entertainment Software Rating Board

Analisar os jogos e verificar padrões que avaliam o sucesso de cada jogo.

Inicialização

In [103]:

```
# Carregando todas as bibliotecas
import pandas as pd
from scipy.stats import levene
from scipy import stats as st
import numpy as np
from math import factorial
import matplotlib.pyplot as plt
```

Carregando os dados

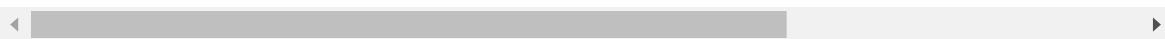
In [104]:

```
#Carregando o arquivo
df= pd.read_csv('/datasets/games.csv')
df
```

Out[104]:

	Name	Platform	Year_of_Release	Genre	NA_sales	EU_sales	JP_sales	Other_Sales	Global_Sales
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	1.01	74.04
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	0.01	35.47
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	0.01	31.24
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	0.01	29.82
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	0.01	20.17
...
16710	Samurai Warriors: Sanada Maru	PS3	2016.0	Action	0.00	0.00	0.01	0.01	0.01
16711	LMA Manager 2007	X360	2006.0	Sports	0.00	0.01	0.00	0.00	0.01
16712	Haitaka no Psychedelica	PSV	2016.0	Adventure	0.00	0.00	0.01	0.01	0.01
16713	Spirits & Spells	GBA	2003.0	Platform	0.01	0.00	0.00	0.00	0.01
16714	Winning Post 8 2016	PSV	2016.0	Simulation	0.00	0.00	0.01	0.01	0.01

16715 rows × 11 columns



Preparar os dados

In [105]:

```
# Renomeação das colunas

df = df.rename(columns={
    'Name': 'name',
    'Platform': 'platform',
    'Year_of_Release': 'year_of_release',
    'Genre': 'genre',
    'NA_sales': 'na_sales',
    'EU_sales': 'eu_sales',
    'JP_sales': 'jp_sales',
    'Other_sales': 'other_sales',
    'Critic_Score': 'critic_score',
    'User_Score': 'user_score',
    'Rating': 'rating'
})

df.columns
```

Out[105]:

```
Index(['name', 'platform', 'year_of_release', 'genre', 'na_sales', 'eu_sales',
       'jp_sales', 'other_sales', 'critic_score', 'user_score', 'rating'],
      dtype='object')
```

In [106]:

```
#Informações do DF
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
 --- 
 0   name            16713 non-null   object  
 1   platform        16715 non-null   object  
 2   year_of_release 16446 non-null   float64 
 3   genre           16713 non-null   object  
 4   na_sales         16715 non-null   float64 
 5   eu_sales         16715 non-null   float64 
 6   jp_sales         16715 non-null   float64 
 7   other_sales     16715 non-null   float64 
 8   critic_score    8137 non-null   float64 
 9   user_score       10014 non-null   object  
 10  rating          9949 non-null   object  
dtypes: float64(6), object(5)
memory usage: 1.4+ MB
```

Valores ausentes

In [107]:

```
# valores ausentes
df.isna().sum()
```

Out[107]:

name	2
platform	0
year_of_release	269
genre	2
na_sales	0
eu_sales	0
jp_sales	0
other_sales	0
critic_score	8578
user_score	6701
rating	6766
dtype: int64	

In [108]:

```
#porcentagem de tbd na coluna user_score:
tbd= 2424
aus=6701
val= 7590

per_tbd= (tbd/(aus+val)) * 100
print('A porcentagem de valor tbd na coluna user_score é:',per_tbd,'%')
```

A porcentagem de valor tbd na coluna user_score é: 16.961724162060037 %

os valores tbd representam cerca de 17% da coluna user_score, vamos avaliar como se o usuário não possui uma avaliação e colocá-lo como ausente

Tratando os valores ausentes

In [109]:

```
#alterando tipos de dados: user_score,year_of_release
df['user_score']= df['user_score'].replace('tbd','NaN')
df['user_score']= df['user_score'].astype('float')
```

valores ausentes nas colunas: name,genre,year_of_release,genre, critic_score,user_score,rating.

name; vamos substituir os valores ausentes por 'unknown', pois desconhecemos os nomes dos ausentes e representam uma parcela muito pequena dos dados na coluna.

genre; vamos substituir os valores ausentes por 'unknown', pois desconhecemos os nomes dos ausentes e representam uma parcela muito pequena dos dados na coluna, não irá comprometer a análise.

year_of_release; vamos substituir pela moda.

critic_score; substituir pela mediana, pois possui valores extremos.

user_score; substituir pela mediana, pois possui valores extremos

rating; substituir por 'unknown', pois não conhecemos as avaliações e representam parte significativa (40%) dos dados e poderíamos comprometer a análise atribuindo algum valor a eles.

In [110]:

```
# name
df['name']= df['name'].fillna('unknown')

# genre
df['genre']= df['genre'].fillna('unknown')

#year_of_release
df['year_of_release']= df['year_of_release'].fillna(df['year_of_release'].mode()[0])

#critic_score
df['critic_score']= df['critic_score'].fillna(df['critic_score'].median())

#user_score
df['user_score']= df['user_score'].fillna(df['user_score'].median())

#rating
df['rating']= df['rating'].fillna('unknown')
```

In [111]:

```
df.isna().sum()
```

Out[111]:

name	0
platform	0
year_of_release	0
genre	0
na_sales	0
eu_sales	0
jp_sales	0
other_sales	0
critic_score	0
user_score	0
rating	0
dtype: int64	

In [112]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16715 entries, 0 to 16714
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   name              16715 non-null   object  
 1   platform          16715 non-null   object  
 2   year_of_release   16715 non-null   float64 
 3   genre              16715 non-null   object  
 4   na_sales           16715 non-null   float64 
 5   eu_sales           16715 non-null   float64 
 6   jp_sales           16715 non-null   float64 
 7   other_sales        16715 non-null   float64 
 8   critic_score       16715 non-null   float64 
 9   user_score          16715 non-null   float64 
 10  rating             16715 non-null   object  
dtypes: float64(7), object(4)
memory usage: 1.4+ MB
```

Valores duplicados

In [113]:

df.duplicated().sum()

Out[113]:

0

In [114]:

df.head()

Out[114]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	8.45
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	0.77
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	3.29
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	2.95
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	1.00



Enriquecer os dados

Criar coluna para Vendas globais

In [115]:

```
# Soma do total de vendas de cada jogo
```

```
df['total_sales'] = df['na_sales'] + df['eu_sales'] + df['jp_sales'] + df['other_sales']
```

In [116]:

```
df.head()
```

Out[116]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	8.45
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	0.77
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	3.29
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	2.95
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	1.00

Análise dos dados

Quantidade de jogos lançados por ano

In [117]:

```
#Estipular quantos jogos foram Lançados por ano
```

```
jogos_ano = df.pivot_table(index='year_of_release', values='name', aggfunc='count').sort_values()
jogos_ano.head()
```

Out[117]:

	year_of_release	name
0	2008.0	1696
1	2009.0	1426
2	2010.0	1255
3	2007.0	1197
4	2011.0	1136

Venda global das plataformas

In [118]:

```
#total de vendas por tipo de plataforma  
vendas_plat=df.pivot_table(index='platform',values='total_sales',aggfunc='sum').sort_values()  
vendas_plat.head(5)
```

Out[118]:

	platform	total_sales
0	PS2	1255.77
1	X360	971.42
2	PS3	939.65
3	Wii	907.51
4	DS	806.12

Vamos assumir as seis maiores vendas totais: -PS2 -X360 -PS3 -WII -DS

Venda global das principais plataformas por ano

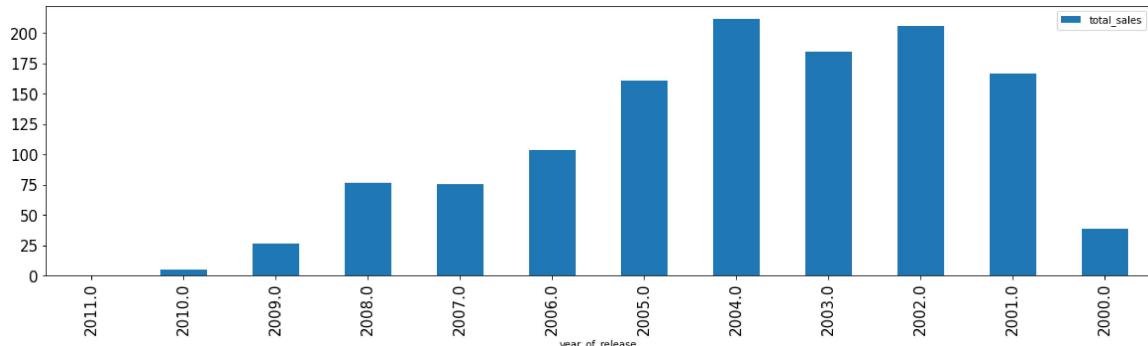
In [119]:

```
#Distribuição por ano de PS2:
```

```
ps2= df[df['platform']=='PS2']
ps2= ps2.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()
ps2.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)
print(ps2)
```

◀ ▶

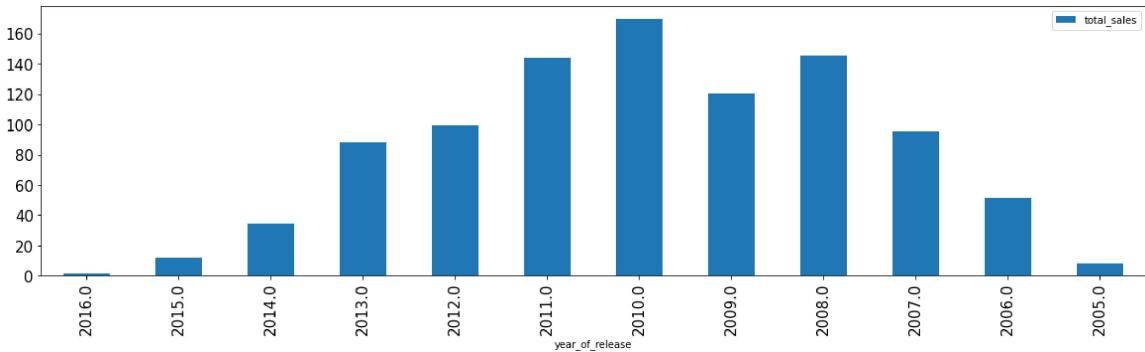
	year_of_release	total_sales
11	2011.0	0.45
10	2010.0	5.64
9	2009.0	26.40
8	2008.0	76.11
7	2007.0	75.99
6	2006.0	103.42
5	2005.0	160.66
4	2004.0	211.81
3	2003.0	184.31
2	2002.0	205.38
1	2001.0	166.43
0	2000.0	39.17



In [120]:

```
#Distribuição por ano de X360:  
x360= df[df['platform']=='X360']  
x360= x360.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_  
x360.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)  
print(x360)
```

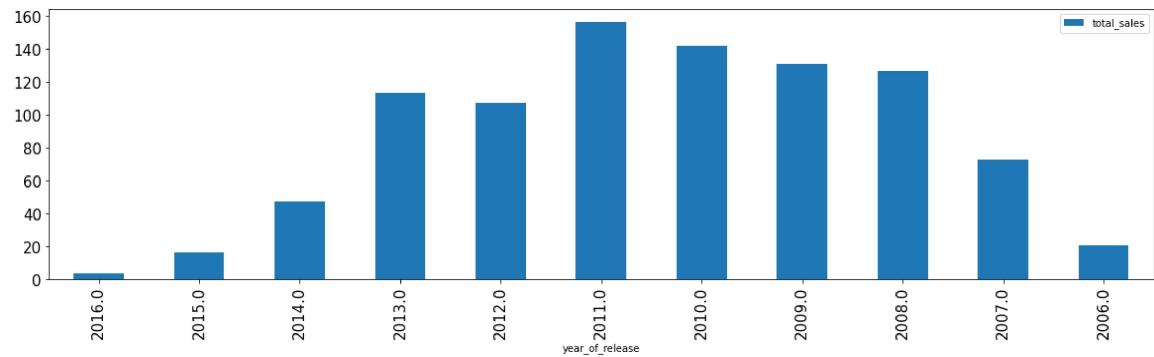
	year_of_release	total_sales
11	2016.0	1.52
10	2015.0	11.96
9	2014.0	34.74
8	2013.0	88.58
7	2012.0	99.74
6	2011.0	143.84
5	2010.0	170.03
4	2009.0	120.29
3	2008.0	145.44
2	2007.0	95.41
1	2006.0	51.62
0	2005.0	8.25



In [121]:

```
#Distribuição por ano de PS3:  
ps3= df[df['platform']=='PS3']  
ps3= ps3.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()  
ps3.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)  
print(ps3)
```

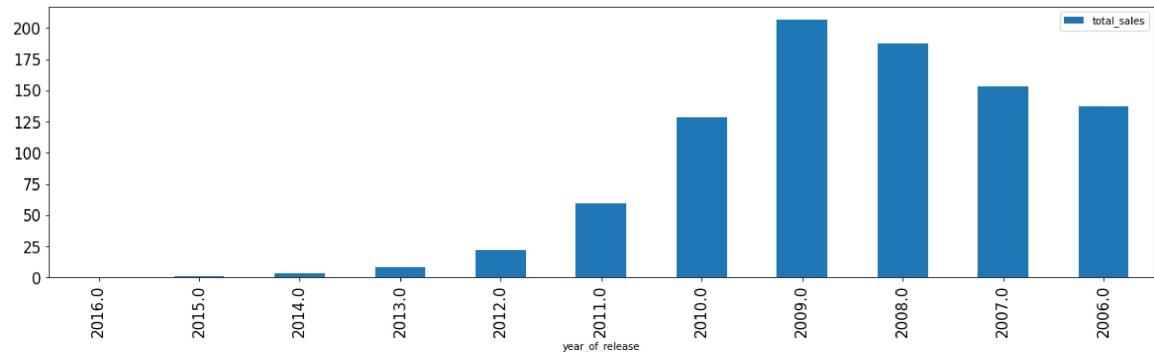
	year_of_release	total_sales
10	2016.0	3.60
9	2015.0	16.82
8	2014.0	47.76
7	2013.0	113.25
6	2012.0	107.36
5	2011.0	156.78
4	2010.0	142.17
3	2009.0	130.93
2	2008.0	126.83
1	2007.0	73.19
0	2006.0	20.96



In [122]:

```
#Distribuição por ano de WII:  
wii= df[df['platform']=='Wii']  
wii= wii.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()  
wii.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)  
print(wii)
```

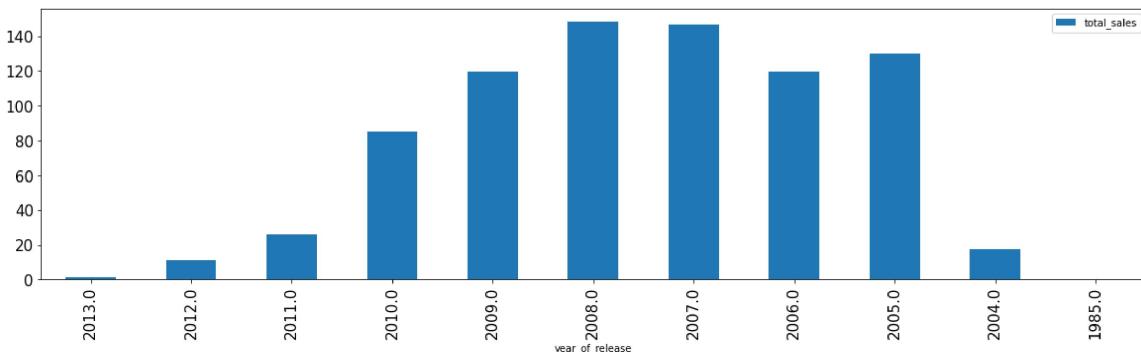
	year_of_release	total_sales
10	2016.0	0.18
9	2015.0	1.14
8	2014.0	3.75
7	2013.0	8.59
6	2012.0	21.71
5	2011.0	59.65
4	2010.0	127.95
3	2009.0	206.97
2	2008.0	187.65
1	2007.0	152.77
0	2006.0	137.15



In [123]:

```
#Distribuição por ano de DS:
ds= df[df['platform']=='DS']
ds= ds.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()
ds.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)
print(ds)
```

	year_of_release	total_sales
10	2013.0	1.54
9	2012.0	11.01
8	2011.0	26.18
7	2010.0	85.02
6	2009.0	119.54
5	2008.0	148.65
4	2007.0	146.94
3	2006.0	119.81
2	2005.0	130.14
1	2004.0	17.27
0	1985.0	0.02



Conclusão da venda global das principais plataformas por ano

CONCLUSÃO

As plataformas DS e PS2, eram bastante populares em anos anteriores, entretanto, atualmente elas perderam espaço e chegam a zerar as vendas. As plataformas Wii e PS3, ainda possuem um pequeno valor total de venda.

As plataformas mais novas começam a aparecer nas vendas cerca de 3 a 5 anos de uso das antigas plataformas, que por sua vez, demoram cerca de 9 anos pra desaparecerem do mercado.

Projeção para o ano consecutivo

Com a nossa análise, podemos perceber que o número de vendas das plataformas que mais venderam ao longo do tempo até hoje estão em queda, e tendem a desaparecer do mercado. Para projeção do próximo ano, vamos olhar para os números a partir de 2014.

In [124]:

```
#Novo dataframe filtrado
df_new= df[df['year_of_release'] >= 2014]
df_new
```

Out[124]:

		name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
31		Call of Duty: Black Ops 3	PS4	2015.0	Shooter	6.03	5.86	0.36	
42		Grand Theft Auto V	PS4	2014.0	Action	3.96	6.31	0.38	
47		Pokemon Omega Ruby/Pokemon Alpha Sapphire	3DS	2014.0	Role-Playing	4.35	3.49	3.10	
77		FIFA 16	PS4	2015.0	Sports	1.12	6.12	0.06	
87		Star Wars Battlefront (2015)	PS4	2015.0	Shooter	2.99	3.49	0.22	
...	
16703		Strawberry Nauts	PSV	2016.0	Adventure	0.00	0.00	0.01	
16707		Aiyoku no Eustia	PSV	2014.0	Misc	0.00	0.00	0.01	
16710		Samurai Warriors: Sanada Maru	PS3	2016.0	Action	0.00	0.00	0.01	
16712		Haitaka no Psychedelica	PSV	2016.0	Adventure	0.00	0.00	0.01	
16714		Winning Post 8 2016	PSV	2016.0	Simulation	0.00	0.00	0.01	

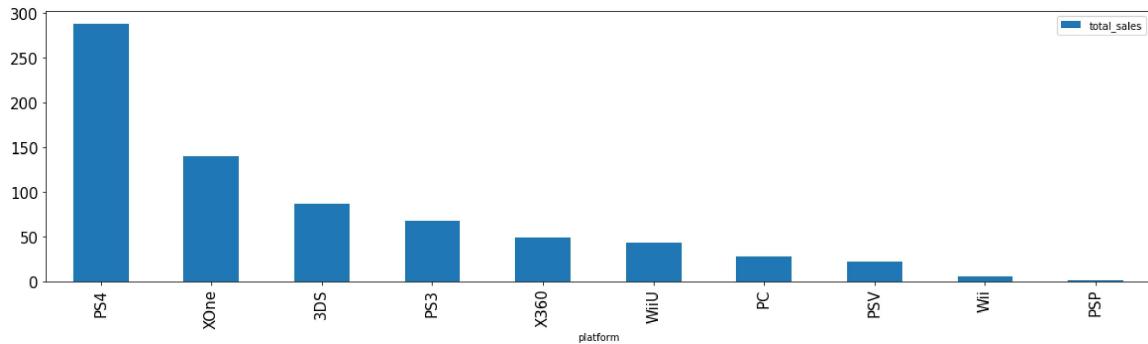
1689 rows × 12 columns



In [125]:

```
#ranking de vendas a partir de 2014
```

```
ranking= df_new.pivot_table(index='platform',values='total_sales',aggfunc='sum').reset_index()
ranking.plot(x='platform',kind='bar',figsize=(20,5),fontsize=15)
plt.show()
print(ranking)
```



```
platform  total_sales
3        PS4      288.15
9       XOne     140.36
0       3DS      86.68
2       PS3      68.18
8       X360     48.22
7       WiiU     42.98
1        PC      27.05
5       PSV      22.40
6        Wii      5.07
4       PSP      0.36
```

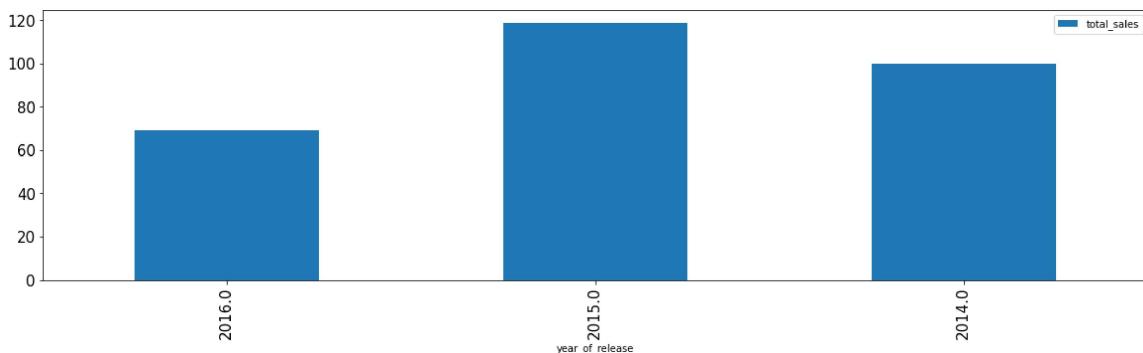
A plataforma PS4 liderar as vendas, seguida de XOne e 3DS.

In [126]:

```
#PS4
ps4= df_new[df_new['platform']=='PS4']
ps4= ps4.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()
ps4.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)
print(ps4)
```

◀ ▶

	year_of_release	total_sales
2	2016.0	69.25
1	2015.0	118.90
0	2014.0	100.00

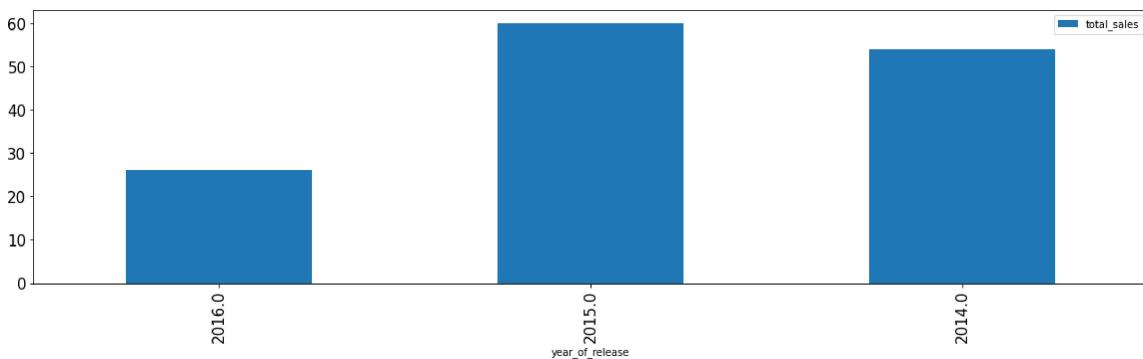


In [127]:

```
#XOne
xone= df_new[df_new['platform']=='XOne']
xone= xone.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()
xone.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)
print(xone)
```

◀ ▶

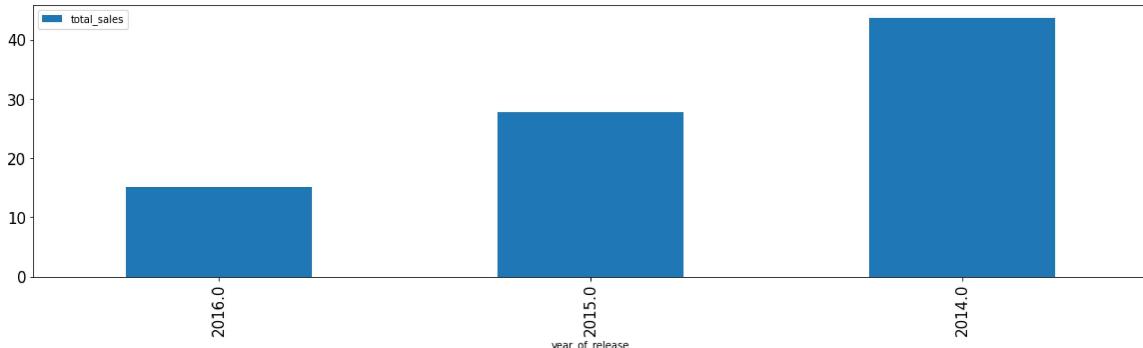
	year_of_release	total_sales
2	2016.0	26.15
1	2015.0	60.14
0	2014.0	54.07



In [128]:

```
#3DS
ds3= df_new[df_new['platform']=='3DS']
ds3= ds3.pivot_table(index='year_of_release',values='total_sales',aggfunc='sum').reset_index()
ds3.plot(x='year_of_release',y='total_sales',kind='bar',figsize=(20,5),fontsize=15)
print(ds3)
```

	year_of_release	total_sales
2	2016.0	15.14
1	2015.0	27.78
0	2014.0	43.76



Conclusão da projeção das plataformas mais promissoras para o ano seguinte

Observa-se que as plataformas PS4 e XOne vem de crescimento de vendas a partir de 2014, enquanto a 3DS vem caindo as vendas. Tomamos que as mais promissoras são a PS4 e XOne.

Análise da venda global de jogos por plataforma

Vamos olhar novamente para o dataframe original(df), para extrair mais informações gerais.

In [129]:

```
#dataframe
df.head()
```

Out[129]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other_sales
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	8.45
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	0.77
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	3.29
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	2.95
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	1.00

In [130]:

#Média de venda global de jogos por plataforma

```
media_jogo_plat= df.pivot_table(index=['name','platform'],values='total_sales',aggfunc='n
.sor
t_values(by='total_sales',ascending=False)
media_jogo_plat
```

Out[130]:

		name	platform	total_sales
16043		Wii Sports	Wii	82.54
13465		Super Mario Bros.	NES	40.24
7999		Mario Kart Wii	Wii	35.52
16045		Wii Sports Resort	Wii	32.77
10571		Pokemon Red/Pokemon Blue	GB	31.38
...	
6817	Katekyoo Hitman Reborn! Kindan no Yami no Delta		Wii	0.01
12743		SoniPro	3DS	0.01
2739	Dark Parables: The Exiled Prince		PC	0.01
11766	SCORE International Baja 1000: The Official Game		PS2	0.00
4933		G1 Jockey 4 2008	PS3	0.00

16710 rows × 3 columns

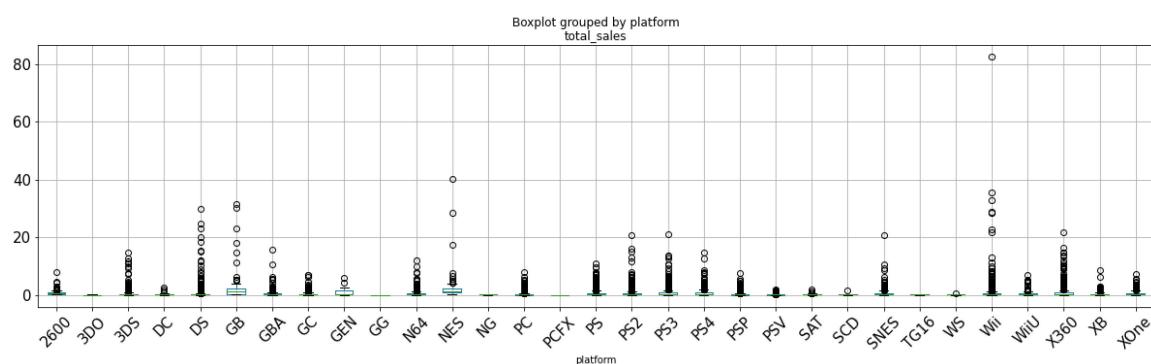
In [131]:

#Diagrama de caixa(média das vendas de jogo por plataforma)

media_jogo_plat.boxplot('total_sales',by='platform',fontsize=15,figsize=(20,5),rot=45)

Out[131]:

<AxesSubplot:title={'center':'total_sales'}, xlabel='platform'>



Várias plataformas possuem jogos que se destacam acima da média e outras não possuem grandes destaques(3DO,DC,GG,NG,PCFX,SCD,TG16,WS). Uma em especial(Wii), onde o jogo Wii Sports é destacado como o pico de maior venda.

Análise da plataforma mais popular

Tomaremos a plataforma Wii como a popular para estudar como as avaliações afetam a venda de uma plataforma:

In [132]:

```
#Filtro para plataforma Wii
df_wii= df[df['platform']=='Wii']
df_wii
```

Out[132]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	
7	Wii Play	Wii	2006.0	Misc	13.96	9.18	2.93	
8	New Super Mario Bros. Wii	Wii	2009.0	Platform	14.44	6.94	4.70	
...
16589	Bomberman	Wii	2008.0	Puzzle	0.00	0.00	0.01	
16626	The Ultimate Battle of the Sexes	Wii	2010.0	Misc	0.01	0.00	0.00	
16664	Mahou Sensei Negima! Neo-Pactio Fight!!	Wii	2007.0	Fighting	0.00	0.00	0.01	
16685	Help Wanted: 50 Wacky Jobs (jp sales)	Wii	2008.0	Simulation	0.00	0.00	0.01	
16686	Yattaman Wii: BikkuriDokkiri Machine de Mou Ra...	Wii	2008.0	Racing	0.00	0.00	0.01	

1320 rows × 12 columns



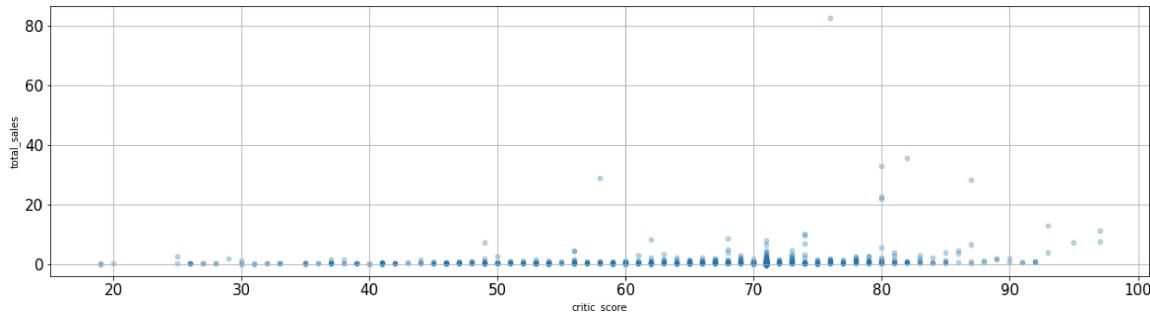
In [133]:

#Gráfico de dispersão da pontuação da crítica

df_wii.plot(x='critic_score',y='total_sales',kind='scatter',figsize=(20,5),fontsize=15,grid=True)

Out[133]:

<AxesSubplot:xlabel='critic_score', ylabel='total_sales'>



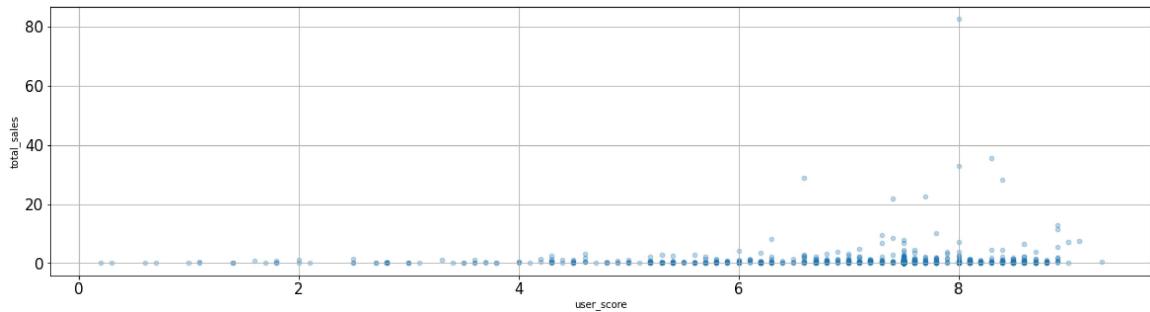
In [134]:

#Gráfico de dispersão da pontuação do usuário

df_wii.plot(x='user_score',y='total_sales',kind='scatter',figsize=(20,5),fontsize=15,grid=True)

Out[134]:

<AxesSubplot:xlabel='user_score', ylabel='total_sales'>



In [135]:

#Correlação entre as avaliações e as vendas

print(df_wii['total_sales'].corr(df_wii['critic_score']))
print(df_wii['total_sales'].corr(df_wii['user_score']))

0.1064946516238445

0.05447292941316128

Conclusão da análise da plataforma mais popular

Podemos dizer que existe uma correlação linear fraca entre as vendas e a pontuação dos críticos, pois a correlação tende mais para zero do que para um. E podemos dizer que não existe correlação entre as vendas e a pontuação dos usuários.

Análise dos Gêneros

In [136]:

```
df
```

Out[136]:

	name	platform	year_of_release	genre	na_sales	eu_sales	jp_sales	other
0	Wii Sports	Wii	2006.0	Sports	41.36	28.96	3.77	
1	Super Mario Bros.	NES	1985.0	Platform	29.08	3.58	6.81	
2	Mario Kart Wii	Wii	2008.0	Racing	15.68	12.76	3.79	
3	Wii Sports Resort	Wii	2009.0	Sports	15.61	10.93	3.28	
4	Pokemon Red/Pokemon Blue	GB	1996.0	Role-Playing	11.27	8.89	10.22	
...
16710	Samurai Warriors: Sanada Maru	PS3	2016.0	Action	0.00	0.00	0.01	
16711	LMA Manager 2007	X360	2006.0	Sports	0.00	0.01	0.00	
16712	Haitaka no Psychedelica	PSV	2016.0	Adventure	0.00	0.00	0.01	
16713	Spirits & Spells	GBA	2003.0	Platform	0.01	0.00	0.00	
16714	Winning Post 8 2016	PSV	2016.0	Simulation	0.00	0.00	0.01	

16715 rows × 12 columns



In [137]:

#Venda de jogos por gênero

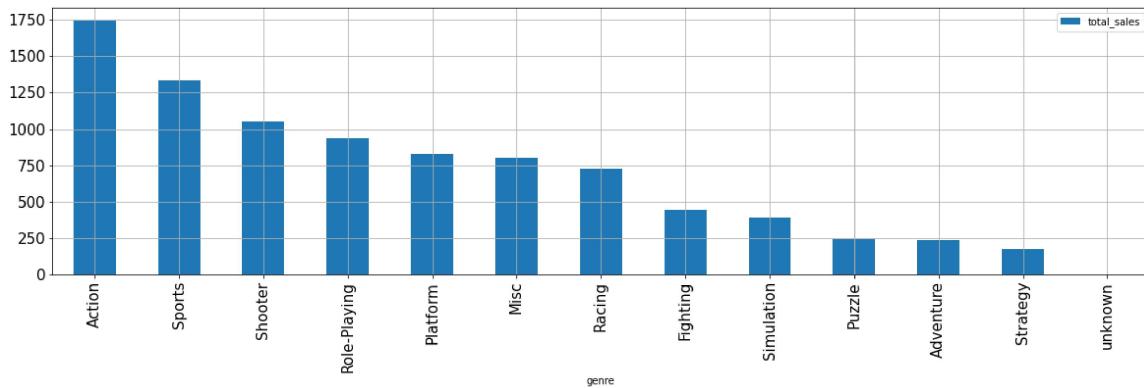
```
jogos_gen = df.pivot_table(index='genre', values='total_sales', aggfunc='sum').reset_index()
print(jogos_gen)
```

```
jogos_gen.plot(x='genre', y='total_sales', kind='bar', grid=True, figsize=(20,5), fontsize=15)
```

	genre	total_sales
0	Action	1744.17
10	Sports	1331.27
8	Shooter	1052.45
7	Role-Playing	934.56
4	Platform	827.77
3	Misc	802.51
6	Racing	728.67
2	Fighting	447.53
9	Simulation	389.97
5	Puzzle	242.57
1	Adventure	237.59
11	Strategy	174.23
12	unknown	2.42

Out[137]:

<AxesSubplot:xlabel='genre'>



Conclusão da análise dos gêneros

Observa-se acima em ordem descrecente os gêneros que geraram maior valor em vendas. Os gêneros Action, Sports e Shooter respectivamente, geraram as maiores valores de vendas global

Perfil de usuário de cada região

As diferenças das quotas de região para região das cinco plataformas principais e dos cinco principais gêneros.

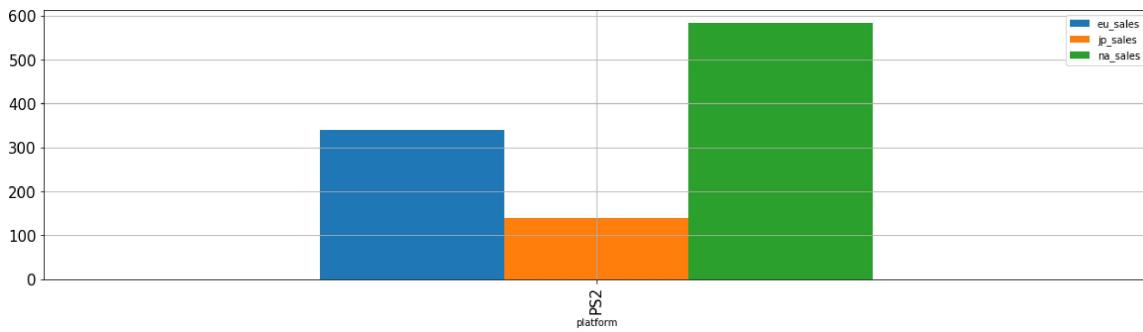
PRINCIPAIS PLATAFORMAS:-PS2 -X360 -PS3 -WII -DS

In [138]:

```
#ps2
ps2=df[df['platform']=='PS2']

ps2= ps2.pivot_table(index='platform',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
ps2.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(ps2)
```

platform	eu_sales	jp_sales	na_sales
PS2	339.29	139.2	583.84

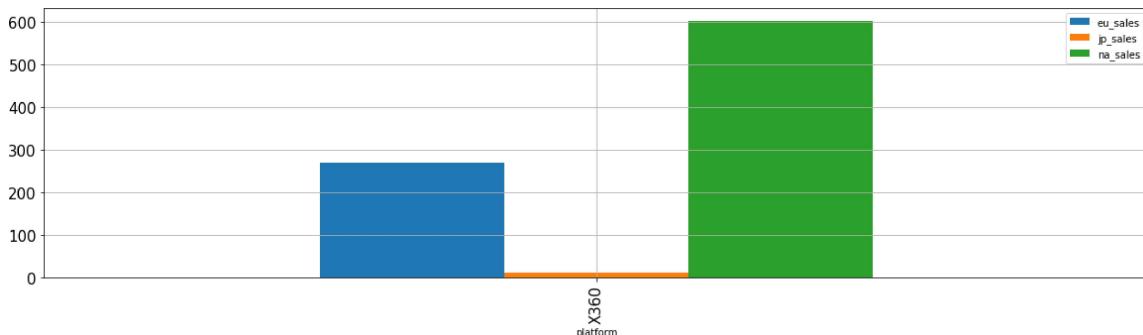


In [139]:

```
#X360
x360=df[df['platform']=='X360']

x360= x360.pivot_table(index='platform',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
x360.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(x360)
```

platform	eu_sales	jp_sales	na_sales
X360	270.76	12.43	602.47

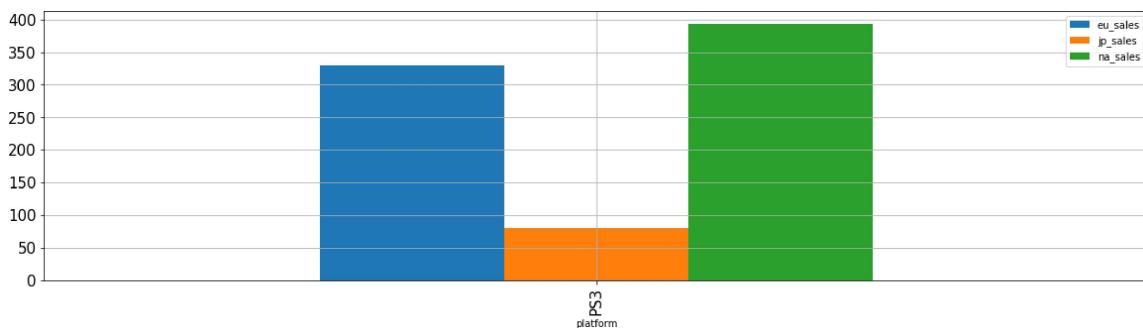


In [140]:

```
#PS3
ps3=df[df['platform']=='PS3']

ps3= ps3.pivot_table(index='platform',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
ps3.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(ps3)
```

	eu_sales	jp_sales	na_sales
platform			
PS3	330.29	80.19	393.49

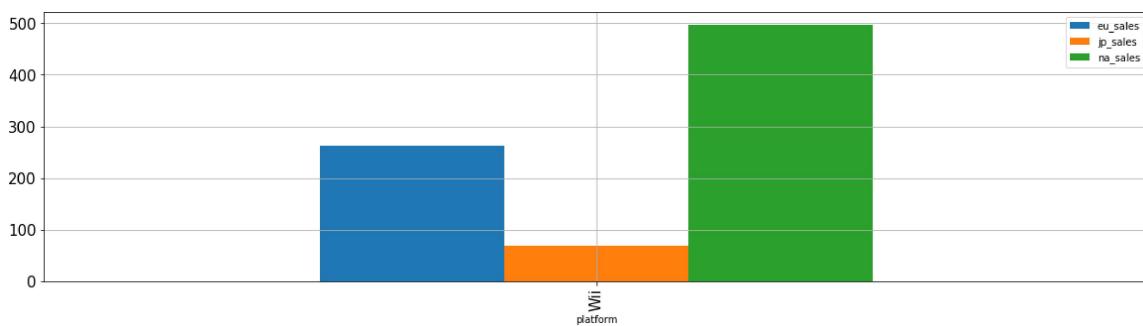


In [141]:

```
#Wii
wii=df[df['platform']=='Wii']

wii= wii.pivot_table(index='platform',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
wii.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(wii)
```

	eu_sales	jp_sales	na_sales
platform			
Wii	262.21	69.33	496.9

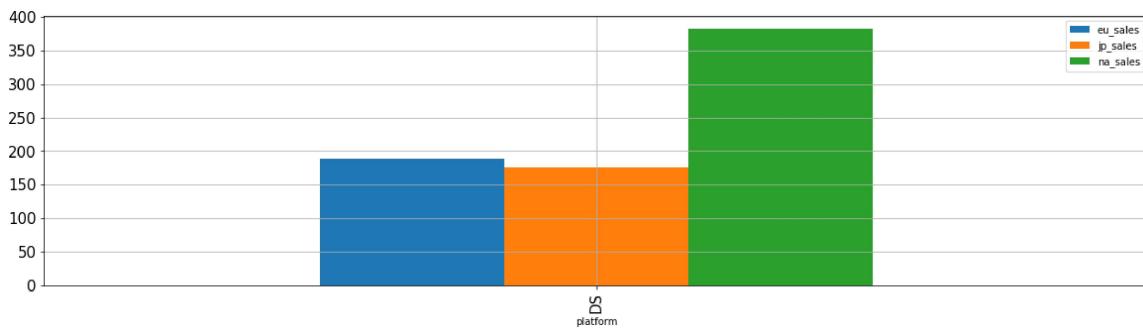


In [142]:

```
#DS
ds=df[df['platform']=='DS']

ds= ds.pivot_table(index='platform',values=['na_sales','eu_sales','jp_sales'],aggfunc='su
ds.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(ds)
```

platform	eu_sales	jp_sales	na_sales
DS	188.89	175.57	382.4



CONCLUSÃO DAS PRINCIPAIS PLATAFORMAS NAS TRÊS REGIÕES

Em todas as plataformas o consumo é maior para os Norte-Americanos! Apenas para a plataforma PS3, a europa se aproxima do consumo dos norte-americanos. O japão possui o menor consumo em todas as plataformas, porém se aproxima dos europeus na plataforma DS.

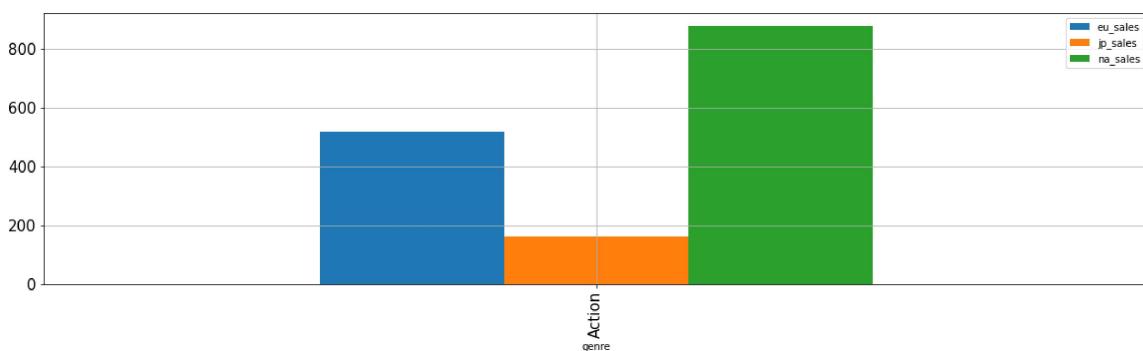
PRINCIPAIS GÊNEROS: - Action -Sports -Shooter -Role-playing -Platform.

In [143]:

```
#Action
action=df[df['genre']=='Action']

action= action.pivot_table(index='genre',values=['na_sales','eu_sales','jp_sales'],aggf
action.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(action)
```

genre	eu_sales	jp_sales	na_sales
Action	519.13	161.43	879.01

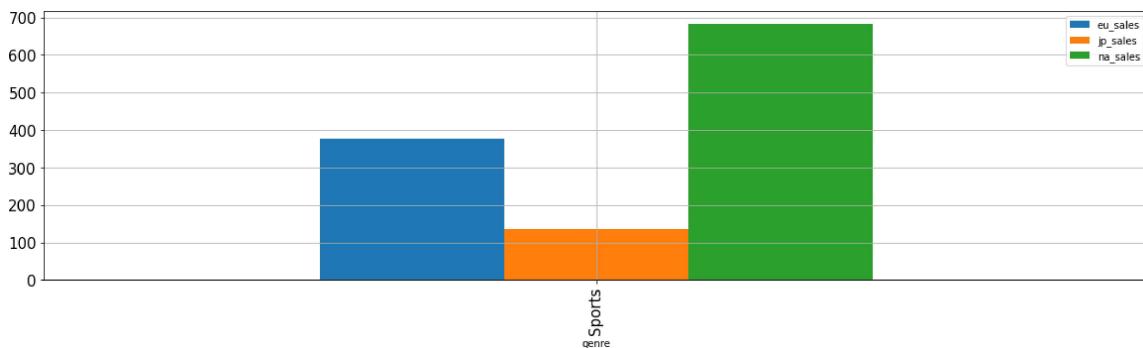


In [144]:

```
#Sports
sports=df[df['genre']=='Sports']

sports=sports.pivot_table(index='genre',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
sports.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(sports)
```

	eu_sales	jp_sales	na_sales
genre			
Sports	376.79	135.54	684.43

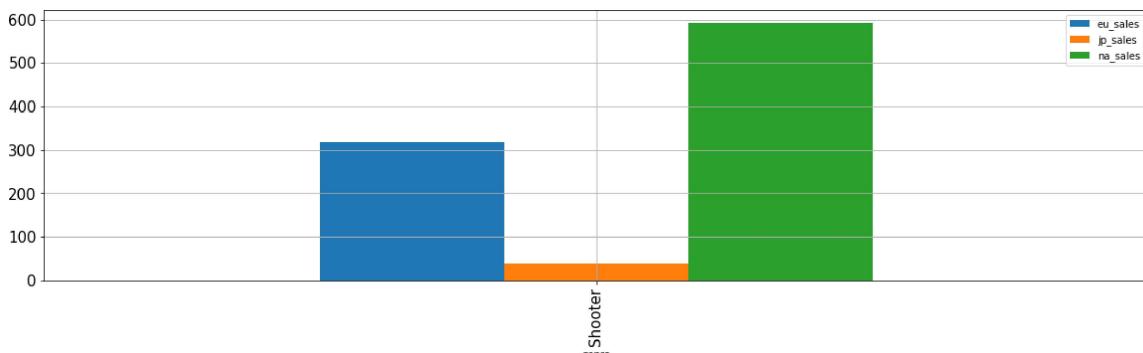


In [145]:

```
#Shooter
shooter=df[df['genre']=='Shooter']

shooter=shooter.pivot_table(index='genre',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
shooter.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(shooter)
```

	eu_sales	jp_sales	na_sales
genre			
Shooter	317.34	38.76	592.24



In [146]:

```
df['genre'].unique()
```

Out[146]:

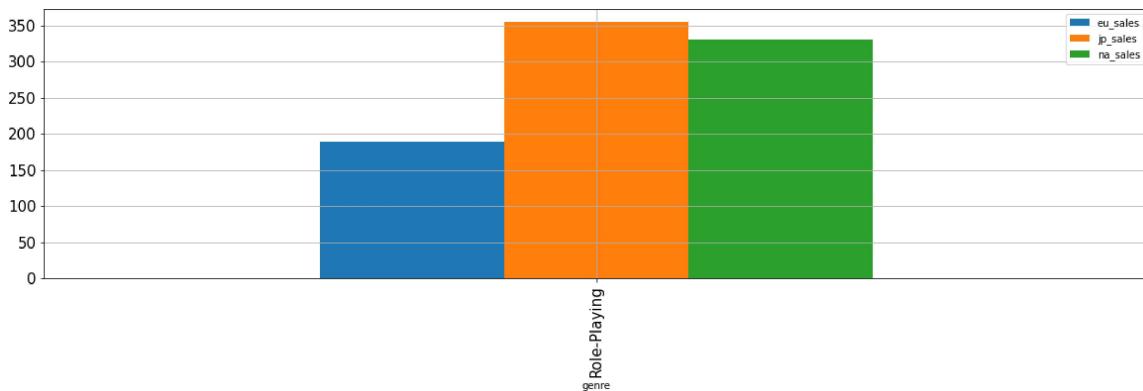
```
array(['Sports', 'Platform', 'Racing', 'Role-Playing', 'Puzzle', 'Misc',
       'Shooter', 'Simulation', 'Action', 'Fighting', 'Adventure',
       'Strategy', 'unknown'], dtype=object)
```

In [147]:

```
#Role_playing
role=df[df['genre']=='Role-Playing']

role= role.pivot_table(index='genre',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
role.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(role)
```

genre	eu_sales	jp_sales	na_sales
Role-Playing	188.71	355.41	330.81

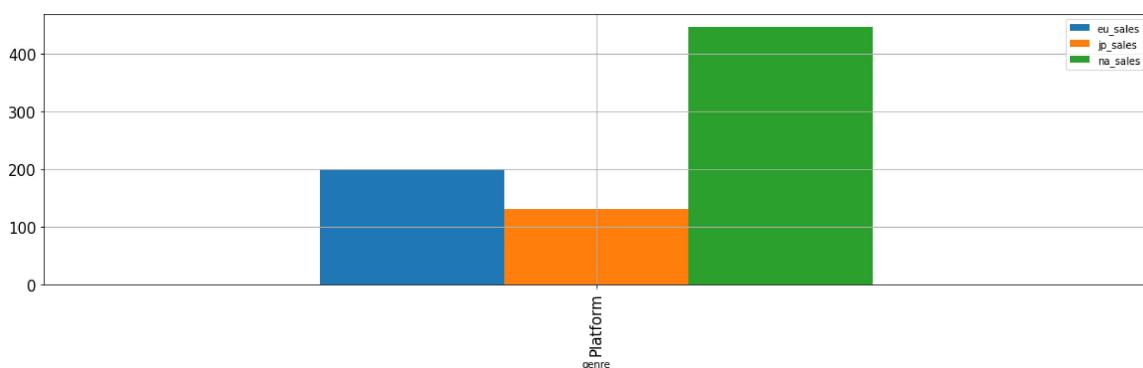


In [148]:

```
#Platform
platform=df[df['genre']=='Platform']

platform= platform.pivot_table(index='genre',values=['na_sales','eu_sales','jp_sales'],aggfunc='sum')
platform.plot(kind='bar',figsize=(20,5),fontsize=15,grid=True)
print(platform)
```

genre	eu_sales	jp_sales	na_sales
Platform	200.35	130.83	445.5



CONCLUSÃO DOS PRINCIPAIS GÊNEROS NAS TRÊS REGIÕES

O comportamento dos principais gêneros é parecido com o comportamento das principais plataformas. Em quase todos os gêneros o consumo dos norte-americanos é superior, exceto no gênero Role-Playing, que o consumo japonês ultrapassa os norte-americanos e os europeus. Em todos os outros gêneros, os europeus ficam abaixo dos norte americanos e acima dos japoneses.

Classificações ERSB

Siglas

E (Everyone- maiores de 6 anos ou livres)/ T (Teen - maiores de 13 anos)/ M (Mature- maiores de 17 anos)/ E10+ (Maiores de 10 anos)/ EC (até 3 anos)/ RP (Em análise)/ K-A (maiores de 6 anos)/ AO (maiores de 18 anos)

In [149]:

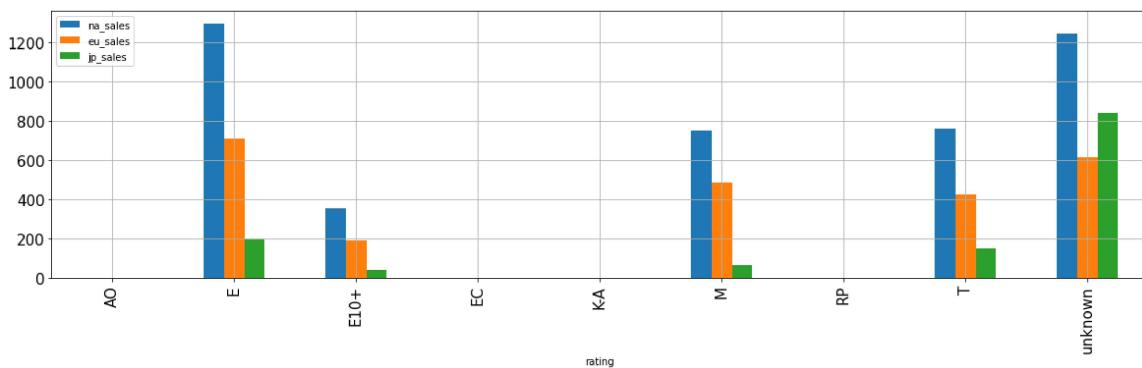
```
# Vendas em regiões por classificação ERSB
rating_na= df.groupby('rating').agg({'na_sales':'sum'}).reset_index()
rating_eu= df.groupby('rating').agg({'eu_sales':'sum'}).reset_index()
rating_jp= df.groupby('rating').agg({'jp_sales':'sum'}).reset_index()

rating_na_eu =rating_na.merge(rating_eu,on='rating',how='outer')
rating_na_eu_jp= rating_na_eu.merge(rating_jp,on='rating',how='outer')
print(rating_na_eu_jp)
rating_na_eu_jp.plot(x='rating',kind='bar',figsize=(20,5),fontsize=15,grid=True)
```

	rating	na_sales	eu_sales	jp_sales
0	AO	1.26	0.61	0.00
1	E	1292.99	710.25	198.11
2	E10+	353.32	188.52	40.20
3	EC	1.53	0.11	0.00
4	K-A	2.56	0.27	1.46
5	M	748.48	483.97	64.24
6	RP	0.00	0.08	0.00
7	T	759.75	427.03	151.40
8	unknown	1242.46	613.83	841.96

Out[149]:

<AxesSubplot:xlabel='rating'>



CONCLUSÃO CLASSIFICAÇÃO ESRB

O consumo para categoria E (maiores de seis anos ou livres) é maior em todas as regiões, com o pico de vendas registrado para os norte-americanos. Os norte-americanos tem o maior consumo em todas as categorias, em seguida vem os europeus que têm o menor consumo que os japoneses apenas na categoria K-A(maiores de seis anos). A venda da categoria E (maiores de seis anos ou livres) é superior em todas as regiões.

Média e variação

Testar Hipóteses Estatísticas

Hipótese 1: As classificações médias dos usuários das plataformas Xbox One e PC são as mesmas.

In [152]:

```
#Avaliando as amostras para definir equal_var (teste de Levene)

xbox=df.query('platform == "XOne"')
pc=df.query('platform == "PC"')

sample_1= xbox['user_score'][xbox['user_score'].notna()].tolist()
sample_2= pc['user_score'][pc['user_score'].notna()].tolist()
alpha=0.05
stat, p = st.levene(sample_1,sample_2)
result_levene = p

if result_levene < alpha:
    print('Rejeita H0, variâncias são diferentes, equal_var = False para o teste')
else:
    print('Não rejeite H0, variâncias não diferem, equal_var = True para o teste')

var_sample= [np.var(x,ddof=1) for x in [sample_1,sample_2]]
print('Sample variances:', var_sample)
```

Não rejeite H0, variâncias não diferem, equal_var = True para o teste
 Sample variances: [1.5895549850235344, 1.881745706983841]

In [153]:

```
#testando as hipóteses:

# H0- As classificações médias dos usuários das plataformas Xbox One e PC são as mesmas.
# H1- As classificações médias dos usuários das plataformas Xbox One e PC são diferentes.

alpha=0.05 #nível crítico de significância estatística
# se for menor que este valor as médias da classificação dos usuários são diferentes.

results= st.ttest_ind(df[df['platform']=='XOne']['user_score'],
                      df[df['platform']=='PC']['user_score'],
                      equal_var=True)
print('p-value:', results.pvalue)

if results.pvalue < alpha:
    print('As classificações médias dos usuários das plataformas Xbox One e PC são diferentes.')
else:
    print('Não podemos rejeitar a hipótese das médias serem iguais')
```

p-value: 0.00010123955107193582
 As classificações médias dos usuários das plataformas Xbox One e PC são diferentes.

Hipótese 2: As classificações médias de usuários para os gêneros Action (ação) e Sports (esportes) são diferentes.

In [579]:

```
#Avaliando as amostras para definir equal_var(teste de Levene)

action= df.query('genre == "Action"')
sports= df.query('genre == "Sports"')

sample_01 = action['user_score'][action['user_score'].notna()].tolist()
sample_02 = sports['user_score'][sports['user_score'].notna()].tolist()
alpha = .05
stat, p = st.levene(sample_01, sample_02)
result_levene = p

if result_levene < alpha:
    print('Rejeita H0, variâncias são diferentes, equal_var = False para o teste')
else:
    print('Não rejeite H0, variâncias não diferem, equal_var = True para o teste')

var_sample_1 = [np.var(x, ddof=1) for x in [sample_1, sample_2]]
print('Sample variances:', var_sample_1)
```

Não rejeite H0, variâncias não diferem, equal_var = True para o teste
 Sample variances: [1.5895549850235344, 1.881745706983841]

In [580]:

```
# Testando as Hipóteses:

# H0 -As classificações médias de usuários para os gêneros Action (ação) e Sports (esport
# H1 -As classificações médias de usuários para os gêneros Action (ação) e Sports (esport

alpha=0.05 #nível crítico de significância estatística
# se for menor que este valor a classificação média dos usuários para os gêneros, são dif

results= st.ttest_ind(df[df['genre']=='Action']['user_score'],
                      df[df['genre']=='Sports']['user_score'],
                      equal_var=True)

print('p-value:', results.pvalue)

if results.pvalue < alpha:
    print('As classificações médias dos usuários para os gêneros são diferentes.')
else:
    print('Não podemos rejeitar a hipótese das médias dos usuários para os gêneros serem iguais')
```

p-value: 0.7131417795320426
 Não podemos rejeitar a hipótese das médias dos usuários para os gêneros serem iguais

Conclusão Geral

Analisamos os dados de vendas da ESRB.

Os dados foram previamente tratados: renomeação das colunas, conversão dos dados, identificação e tratamento dos valores ausentes e também não identificados, os dados foram enriquecidos com o cálculo da venda global para utilizarmos na análise.

Dados tratados, estamos prontos para análise: Foi verificado a venda global das principais plataformas por ano, a projeção das plataformas mais promissoras para o ano seguinte, pegamos a plataforma mais popular para verificar a correlação entre as vendas global e as avaliações dos críticos e dos usuários, verificamos quais gêneros geram maiores vendas em todas as regiões.

Análise do Perfil de usuário de cada região: Realiza-se uma leitura sobre as vendas das principais plataformas e dos principais gêneros nas três regiões. E calculamos o consumo de cada categoria da classificação ESRB por cada uma das três regiões.

Análise sobre a realização dos testes de hipóteses. Dois testes foram realizados:

teste 1= As classificações médias dos usuários das plataformas Xbox One e PC são as mesmas. Realiza-se o teste de levene para estipular o valor do parâmetro equal_var. observação: Estipula-se 5% de significância estatística. Tomamos H0 como a igualdade entre as médias. Descobre-se que, as classificações médias dos usuários das plataformas Xbox One e PC são diferentes.

teste 2= As classificações médias de usuários para os gêneros Action (ação) e Sports (esportes) são diferentes. Realiza-se o teste de levene para estipular o valor do parâmetro equal_var. Observação: Estipula-se 5% de significância estatística. Tomamos H0 como a igualdade entre as médias. Descobre-se que, não podemos rejeitar a hipótese das médias dos usuários para os gêneros serem iguais.