

Esquilos Berrantes (ESB)

- Lucas Jacinto Gonçalves- 240013
- Leonardo Novaes do Nascimento- 220142
- Daniel Mendes dos Santos- 214752

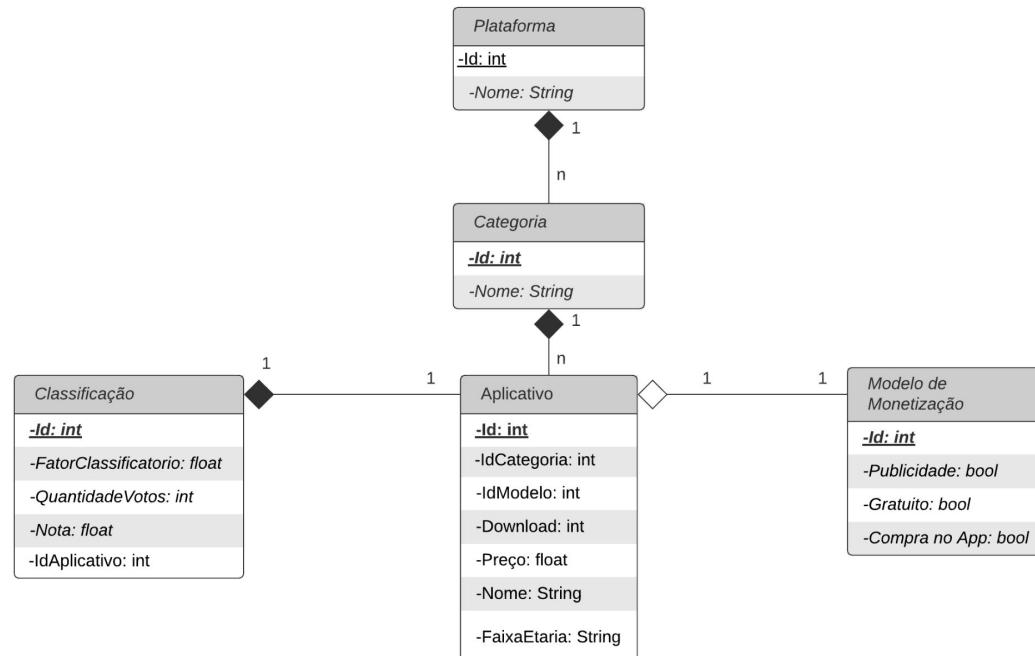


Descrição e Motivação do Dataset

- Android e iOS são as duas maiores plataformas mobile atualmente.
- A partir da análise dos datasets será possível determinar, por exemplo, em qual plataforma é mais lucrativo para determinada empresa anunciar seus produtos.
- Analisar o desempenho de um mesmo aplicativo em ambas as plataformas, com relação a classificação nas lojas, além da relação de qualidade de um aplicativo gratuito com sua versão paga.

Modelo Conceitual

Diagrama UML



Modelo Lógico Tabular

MODELO LÓGICO BANDO DE DADOS

Plataforma		Categoria	
Id	Nome	Id	Nome

Aplicativo							
Id	IdCategoria	IdModelo	IdPlataforma	Download	Preço	Nome	FaixaEtaria

Classificação				
Id	IdAplicativo	FatorClassificatorio	QuantidadeVotos	Nota

Modelo de Monetização			
Id	Publicidade	Gratuito	CompraNoApp

Plataforma (**Id**, Nome)

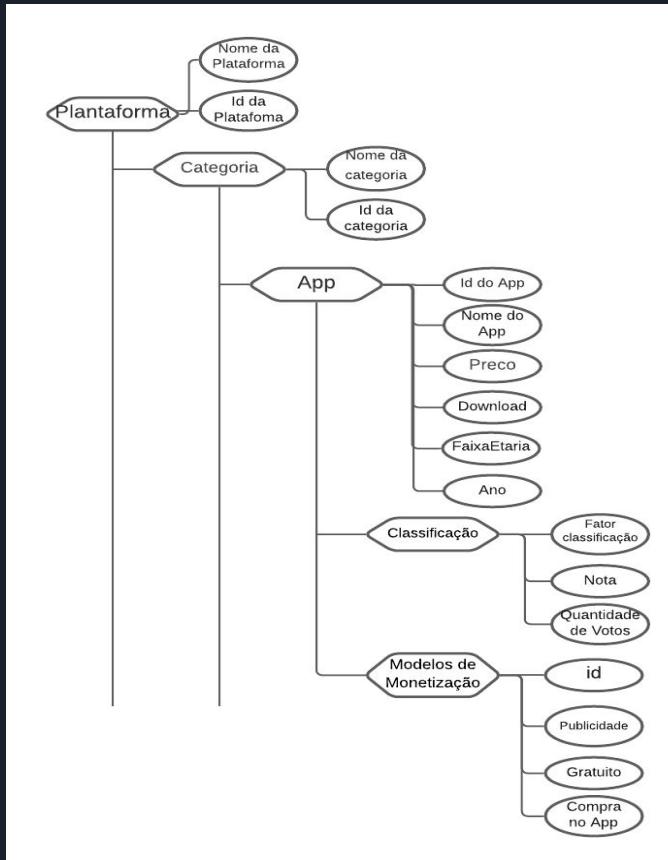
Categoria(**Id**, Nome)

Aplicativo (**Id**, IdCategoria, IdModelo, IdPlataforma, Download, Preço, Nome, FaixaEtaria)

Classificação (**Id**, IdAplicativo, FatorClassificatorio, QuantidadeVotos, Nota)

Modelo de Monetização (**Id**, Publicidade, Gratuito, CompraNoApp)

Modelo Lógico de Documentos





Perguntas de Pesquisa/Análise

- 
- 01 Os conteúdos mais bem avaliados e a relação com as categorias as quais eles pertencem
 - 02 Relacionando aplicativos pagos com aplicativos gratuitos.
 - 03 Levando em consideração a faixa etária e a plataforma no qual o aplicativo pertence, qual a categoria de aplicativo mais baixado por cada público-alvo ?

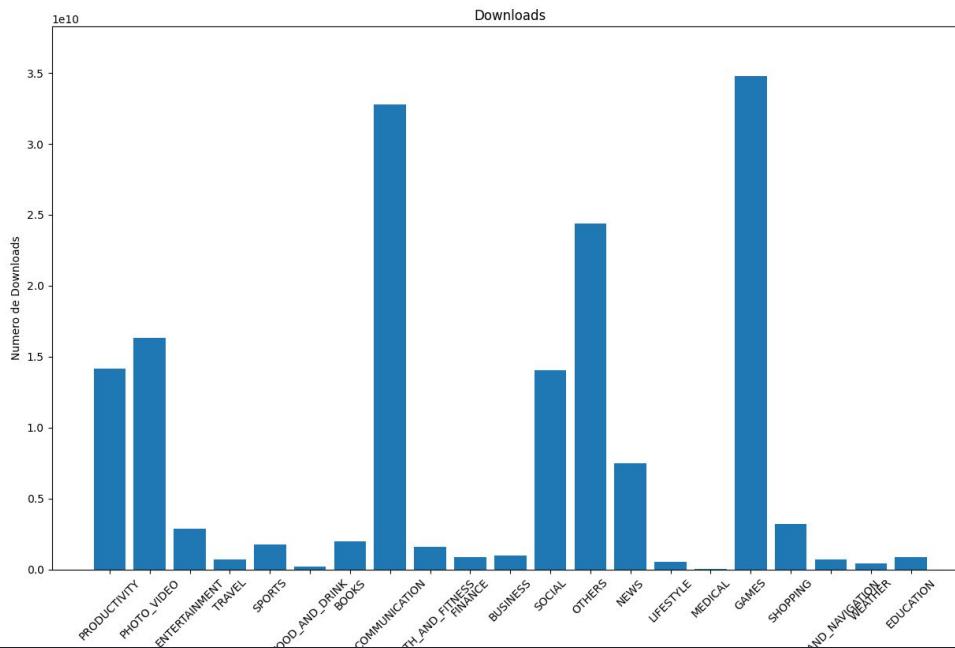


Pergunta 1: Os conteúdos mais bem avaliados e a relação com as categorias as quais eles pertencem

1. WhatsApp Messenger
2. Instagram
3. Messenger Text and Video Chat for Free
4. Clash of Clans
5. Clean Master- Space Cleaner & Antivirus
6. Subway Surfers
7. YouTube
8. Security Master - Antivirus, VPN, AppLock, Booster
9. Clash Royale
10. Candy Crush Saga

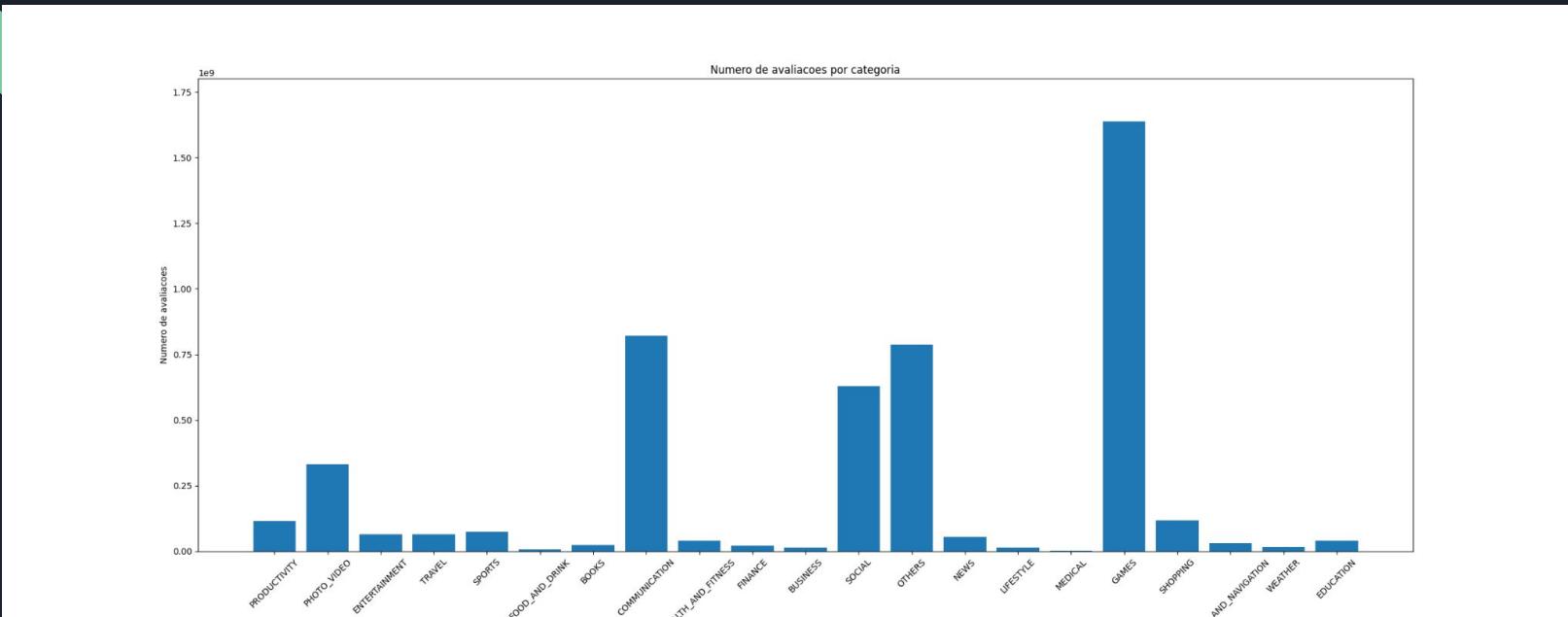
```
SELECT DISTINCT top10.nome from (
SELECT Aplicativo.nome, Class.fator, Categoria.Nome as categoria, Aplicativo.faixa_etaria, Aplicativo.download, Class.quantidadeVotos
FROM app_mc536.Aplicativo, app_mc536.Class, app_mc536.Categoria
WHERE Aplicativo.id = Class.idAplicativo AND Aplicativo.idCategoria = Categoria.id
ORDER BY Class.fator DESC, Class.quantidadeVotos DESC, Class.nota DESC
LIMIT 1, 100
)top10
LIMIT 1, 10;
```

Pergunta 1: Os conteúdos mais bem avaliados e a relação com as categorias as quais eles pertencem



```
SELECT Categoria.nome AS Categorias, SUM(Apps_2019.download) AS Downloads
FROM (SELECT Aplicativo.id, Aplicativo.nome, Aplicativo.Idcategoria, Aplicativo.download, Class.nota, Class.quantidadeVotos, Class.fator
FROM Aplicativo, Class
WHERE Aplicativo.ano = 2019 AND Aplicativo.id = Class.idAplicativo
AND Aplicativo.idPlataforma = 1
) Apps_2019, Categoria
WHERE Apps_2019.IdCategoria = Categoria.id
GROUP BY Apps_2019.IdCategoria;
```

Pergunta 1: Os conteúdos mais bem avaliados e a relação com as categorias as quais eles pertencem



```
SELECT categoria.nome, SUM(Class.quantidadeVotos)
FROM Aplicativo, categoria, Class
WHERE Aplicativo.idCategoria = categoria.id AND Class.idAplicativo
= Aplicativo.id
GROUP BY categoria.nome;
```

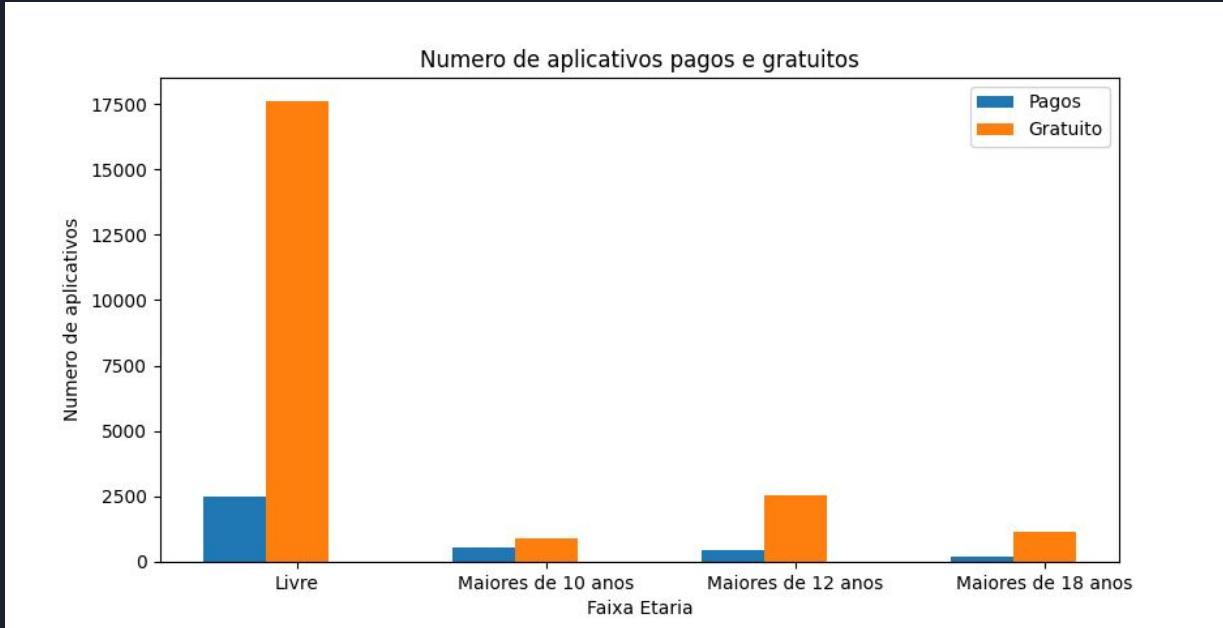


Pergunta 1:

Os conteúdos mais bem avaliados e a relação com as categorias as quais eles pertencem

- Neste banco de dados, podemos encontrar os aplicativos mais bem rentáveis através do fator classificação. Esse fator é calculado a partir da nota e das quantidade de avaliações.
- Além disso podemos relacionar números de downloads, número de avaliações e fator classificação. Além disso podemos encontrar os aplicativos mais bem avaliados para cada tipo de categoria
- (Aqui podemos usar o modelo de documentos já que nem todos os aplicativos possui todas informações conhecidas porém compartilham uma mesma estrutura)

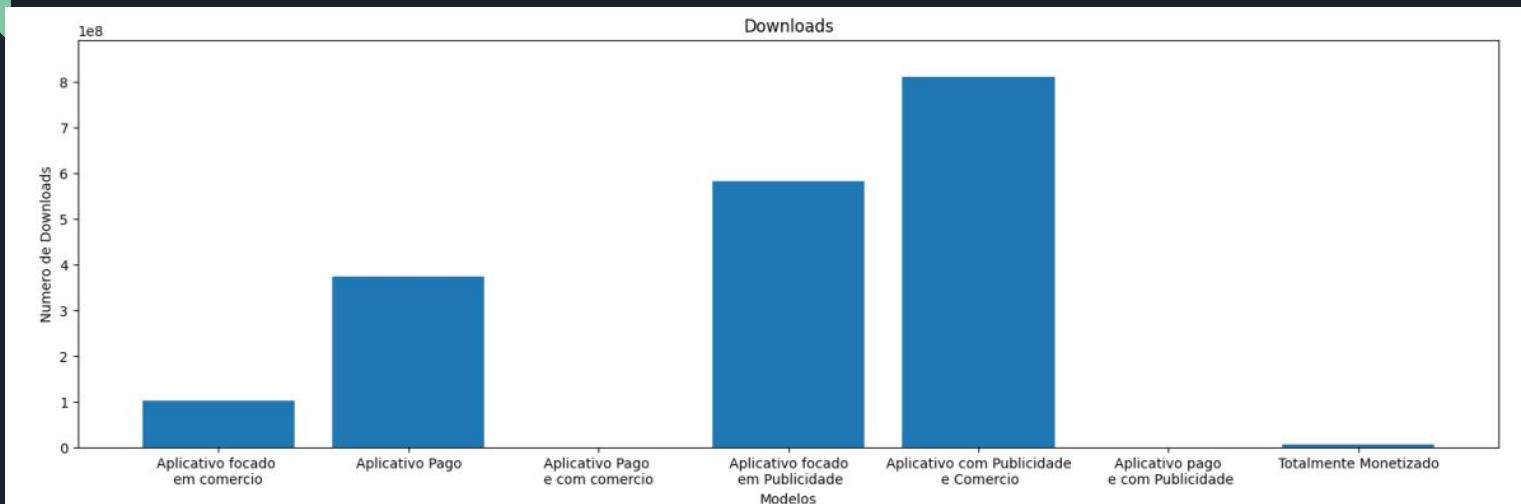
Pergunta 2: Relacionando aplicativos pagos com aplicativos gratuitos.



```
SELECT Aplicativo.faixa_etaria, COUNT(*)  
FROM Aplicativo  
WHERE Aplicativo.preco <> 0  
GROUP BY Aplicativo.faixa_etaria;
```

```
SELECT Aplicativo.faixa_etaria, COUNT(*)  
FROM Aplicativo  
WHERE Aplicativo.preco = 0  
GROUP BY Aplicativo.faixa_etaria;
```

Pergunta 2: Relacionando aplicativos pagos com aplicativos gratuitos.



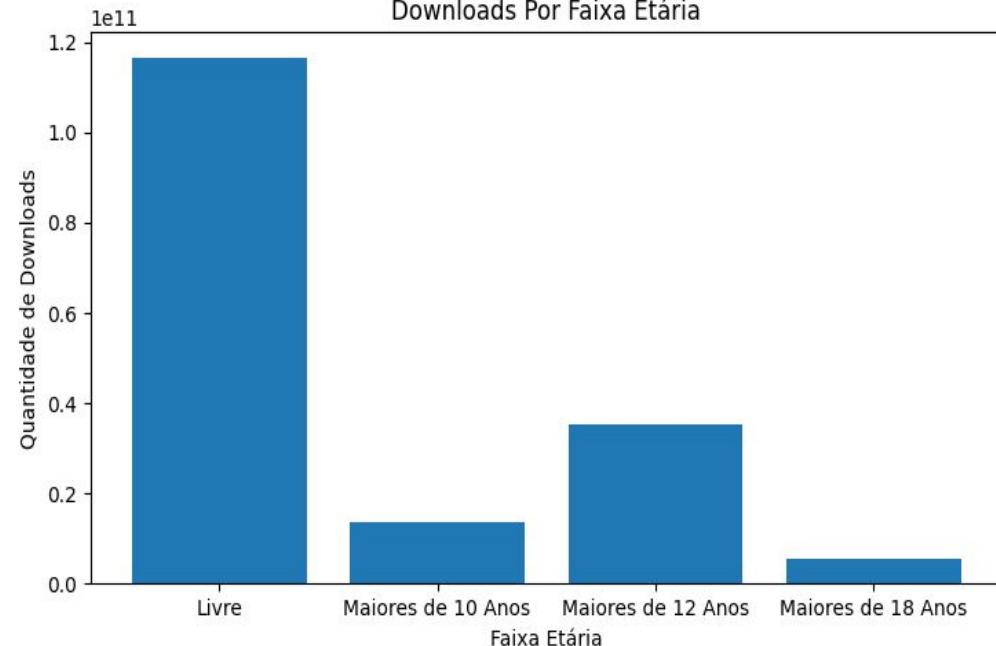
```
SELECT ModMon.Tipo AS Modelo,  
SUM(Ap.download) AS Downloads  
FROM Aplicativo Ap, ModMon  
Where Ap.idModelo = ModMon.id  
GROUP BY ModMon.Tipo;
```



Pergunta 2: Relacionando aplicativos pagos com aplicativos gratuitos.

- Anteriormente existia a categoria “Sem monetização”, porém ela foi excluída por ser muito grande.
- De acordo com o gráfico de Downloads por Modelos, os modelos de monetização como: aplicativo com suporte a publicidade e compras no app (comércio), aplicativo focado em publicidade e aplicativos pagos são os modelos de negócios mais rentáveis atualmente.
- Aplicativos focados em comércio (compras no app) são em torno de 6x menos lucrativos quando em comparação com os aplicativos focados em publicidade e cerca de 8x menos lucrativos quando comparados a aplicativos que possibilitam o suporte a publicidade e comércio (compras no app).

Pergunta 3: Levando em consideração a faixa etária e a plataforma no qual o aplicativo pertence, qual a categoria de aplicativo mais baixado por cada público-alvo ?



```
SELECT A.faixa_etaria, SUM(A.download)
FROM Aplicativo A
GROUP BY A.faixa_etaria;
"""
```



Pergunta 3: Levando em consideração a faixa etária e a plataforma no qual o aplicativo pertence, qual a categoria de aplicativo mais baixado por cada público-alvo ?

- De acordo com o gráfico de Downloads por Faixa Etária, a Faixa Etária Livre possui a maior quantidade de downloads, com mais de 1.0×10^{11} downloads, seguido pela categoria de Maiores de 12 anos com 0.4×10^{11} downloads.
- A Faixa Etária com a menor quantidade de downloads é para o público com idade maior do que 18 anos.
- A Faixa Etária com indicação Livre devido a ser aberto ao público geral permite, por exemplo, que uma empresa direcione publicidade a esses apps, desde que ele possua suporte a isso.
- Mesmo na Faixa Etária Livre ainda é possível ver maior relevância em relação aos outras Faixas

Conexão com MySQL

```
1 import mysql.connector
2 from mysql.connector import Error
3 def create_connection(host_name, user_name, user_password, db_name):
4     connection = None
5     try:
6         connection = mysql.connector.connect(
7             host=host_name,
8             user=user_name,
9             passwd=user_password,
10            database=db_name,
11            )
12         print("Connection to MySQL DB successful")
13     except Error as e:
14         print(f"The error '{e}' occurred")
15     return connection
16
17
18 connection = create_connection("127.0.0.1", "root", "root123", "app_mc536")
```

Criação dos Bancos de Dados

```
try:
    for x in array:
        query = ("INSERT INTO Categoria (Nome) values('{}');".format(x))
        cursor.execute(query)
    query = ("INSERT INTO Plataforma (Nome) values('{}');".format("Android"))
    cursor.execute(query)
    query = ("INSERT INTO Plataforma (Nome) values('{}');".format("Ios"))
    cursor.execute(query)

    connection.commit()
    print("Queries executada!")

except Exception as e:
    print(f"The error '{e}' occurred")

def insert_modelos_database(connection, array):
    cursor = connection.cursor()
    try:
        for x in array:
            query = ("INSERT INTO ModMon (Publicidade,Gratis,CompraApp,Tipo) values({}, {}, {}, '{}');".format(x[0],x[1],x[2],x[3]))
            cursor.execute(query)
    connection.commit()
    print("Queries executada!")

except Exception as e:
    print(f"The error '{e}' occurred")

queries=[  
"CREATE TABLE Plataforma(id int NOT NULL AUTO_INCREMENT PRIMARY KEY,Nome varchar(20) NOT NULL);",  
"CREATE TABLE Categoria(id int NOT NULL AUTO_INCREMENT PRIMARY KEY,Nome varchar(20) NOT NULL);",  
"CREATE TABLE ModMon(id int NOT NULL AUTO_INCREMENT PRIMARY KEY,Publicidade int,Gratis int,CompraApp int,Tipo varchar(80));",  
"CREATE TABLE Aplicativo(id int NOT NULL AUTO_INCREMENT PRIMARY KEY,Nome varchar(255) NOT NULL,preco DECIMAL(5,2),ano int, download int,faixa_etaria varchar(20)",  
"CREATE TABLE Class(id int NOT NULL AUTO_INCREMENT PRIMARY KEY,QuantidadeVotos int,fator DECIMAL(5,2),nota DECIMAL(5,2),idAplicativo int,FOREIGN KEY (idAplicati  
]
```

Dicionário de Categorias para cada plataforma

```
59 id_categorias={  
60     "PRODUCTIVITY": 0,  
61     "PHOTO_VIDEO": 1,  
62     "ENTERTAINMENT": 2,  
63     "TRAVEL": 3,  
64     "SPORTS": 4,  
65     "FOOD_AND_DRINK": 5,  
66     "BOOKS": 6,  
67     "COMMUNICATION": 7,  
68     "HEALTH_AND_FITNESS": 8,  
69     "FINANCE": 9,  
70     "BUSINESS": 10,  
71     "SOCIAL": 11,  
72     "OTHERS": 12,  
73     "NEWS": 13,  
74     "LIFESTYLE": 14,  
75     "MEDICAL": 15,  
76     "GAMES": 16,  
77     "SHOPPING": 17,  
78     "MAPS_AND_NAVIGATION": 18,  
79     "WEATHER": 19,  
80     "EDUCATION": 20  
81 }
```

```
categorias_conv_apple={  
    "Productivity":"PRODUCTIVITY",  
    "Photo & Video":"PHOTO_VIDEO",  
    "Entertainment":"ENTERTAINMENT",  
    "Travel":"TRAVEL",  
    "Sports":"SPORTS",  
    "Food & Drink":"FOOD_AND_DRINK",  
    "Book":"BOOKS",  
    "Music":"COMMUNICATION",  
    "Health & Fitness":"HEALTH_AND_FITNESS",  
    "Finance":"FINANCE",  
    "Business":"BUSINESS",  
    "Social Networking":"SOCIAL",  
    "Utilities":"OTHERS",  
    "News":"NEWS",  
    "Lifestyle":"LIFESTYLE",  
    "Medical":"MEDICAL",  
    "Games":"GAMES",  
    "Catalogs":"OTHERS",  
    "Shopping":"SHOPPING",  
    "Navigation":"MAPS_AND_NAVIGATION",  
    "Reference":"OTHERS",  
    "Weather":"WEATHER",  
    "Education":"EDUCATION",  
}
```

Transformação de dados para Bancos de Dados grandes

- 4 horas para poder inserir todos os dados trabalhados.
- Sendo a complexidade em ordem $O(n)$
- Seriam necessários quase duas semanas para poder inserir todos os dados

```
def publico(case):
    if case == "4+":
        return "Livre"
    elif case == "9+":
        return "Maiores de 10 anos"
    elif case == "12+":
        return "Maiores de 12 anos"
    elif case == "17+":
        return "Maiores de 18 anos"
    else:
        return "Livre"

apple_csv_file="..../arquivos_csv/AppleStore.csv"
```

```
} def publico(case):
    if case == "Everyone":
        return "Livre"
    elif case == "Everyone 10+":
        return "Maiores de 10 anos"
    elif case == "Teen":
        return "Maiores de 12 anos"
    elif case == "Mature 17+":
        return "Maiores de 18 anos"
    else:
        return "Livre"

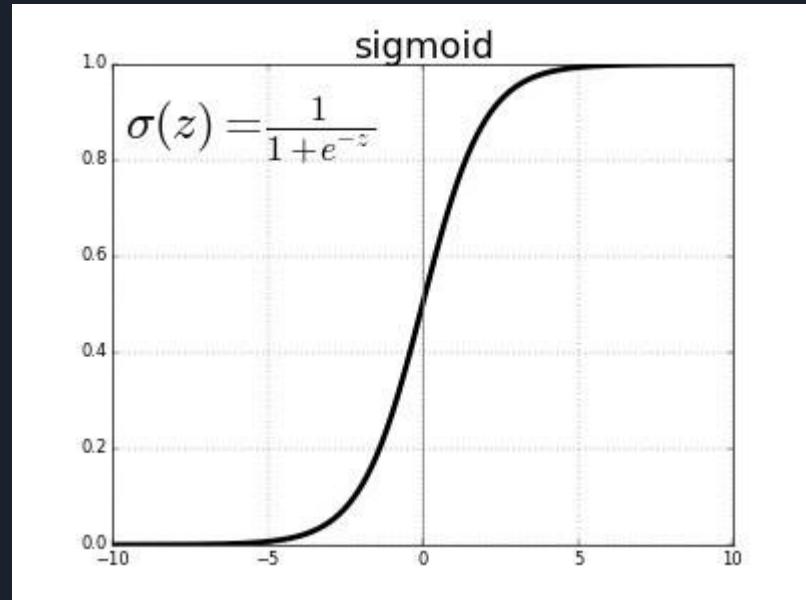
google_2019_csv_file="..../arquivos_csv/googleplaystore.csv"
```

size: 25924

Transformação de dados para Bancos de Dados grandes

```
def publico(case):
    if case == "Everyone":
        return "Livre"
    elif case == "Everyone 10+":
        return "Maiores de 10 anos"
    elif case == "Teen":
        return "Maiores de 12 anos"
    elif case == "Mature 17+":
        return "Maiores de 18 anos"
    else:
        return "Livre"
```

```
google_2021_csv_file="..../arquivos_csv/Google-Playstore.csv"
```





Fonte de dados e operações de preparo no dataset

As fontes de dados utilizados foi do site Kaggle.

<https://www.kaggle.com/gauthamp10/google-playstore-apps>

<https://www.kaggle.com/ramamet4/app-store-apple-data-set-10k-apps>

<https://www.kaggle.com/lava18/google-play-store-apps>

As operações realizadas nos datasets do Android e iOS foram agregação das tabelas, tratamento de informações e transformações.

Obrigado!

