

Presentación sobre el Proyecto:

# "IMDB MOVIES RDF"

Equipo 1:

Arturo Kullmer

Camila Salas

Javiera Labrín



# INTRODUCCIÓN

*mmmm*

Se utilizará un dataset que contiene un top 1000 de películas, el cual se convertirá en un grafo RDF, agregando información utilizando Wikidata.  
Finalmente se harán consultas SPARQL.



# DATASET



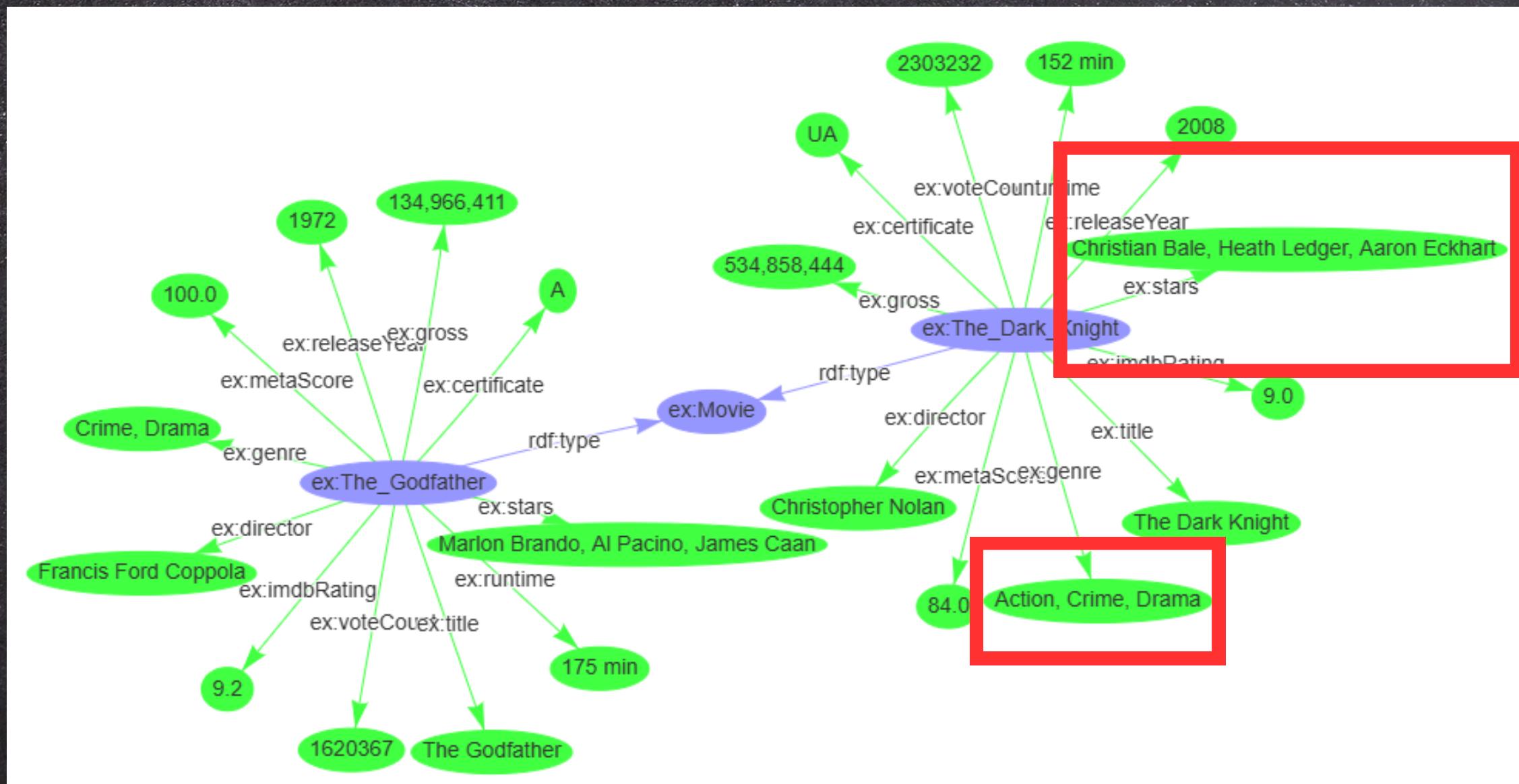
Está compuesto por:

- Nombre de la película
- Año de estreno
- Certificado obtenido
- Duración
- Género
- Rating
- Puntaje
- Nombre del director
- Estrellas (Actores/Actrices)
- Número de votos
- Dinero ganado por la película

# IMPLEMENTACIÓN

Los pasos a seguir para obtener la estructura final son:

1. Convertir el archivo CSV a RDF utilizando la librería de Python “tarql”

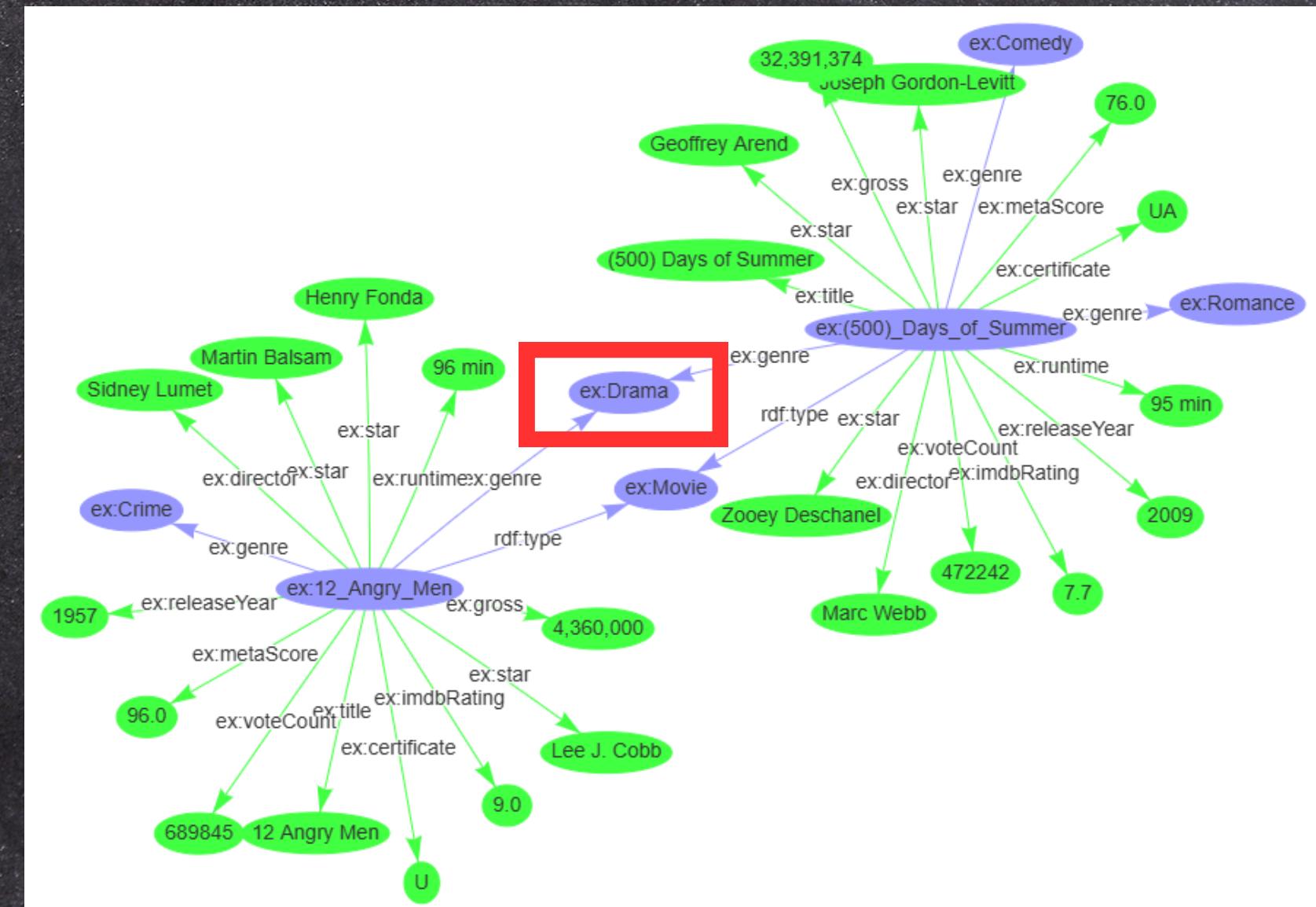


Ejemplo de 2 películas (RDF Playground)

# IMPLEMENTACIÓN

Los pasos a seguir para obtener la estructura final son:

2. Adecuar los datos de género y estrellas a un formato más conveniente para un grafo:



Ejemplo de 2 películas (RDF Playground)

# IMPLEMENTACIÓN



Los pasos a seguir para obtener la estructura final son:

## 3. Enriquecer los datos con la API de wikidata:

- **IMDB ID: identificador IMDB**
- **Capital cost: costo de producción**
- **Narrative location: lugar de ambientación**
- **Production company: compañía que produce la película**

```
url = "https://query.wikidata.org/sparql"

def search_movie_on_wikidata(title, year):
    # Consulta SPARQL para buscar una película
    query = """
    SELECT DISTINCT ?movie ?imdb_id ?capital_cost ?narrative_location ?production_company
    WHERE {
        ?movie wdt:P31 wd:Q11424 ;
               wdt:P1476 ?title ;
               wdt:P577 ?release_date ;
               rdfs:label ?label ;
               wdt:P345 ?imdb_id .
    """
    OPTIONAL { ?movie wdt:P2130 ?capital_cost }
    OPTIONAL { ?movie wdt:P840 ?narrative_location }
    FILTER(lang(?narrative_location) = "es")
    OPTIONAL { ?movie wdt:P272? ?production_company }
    FILTER(lang(?production_company) = "es")
    FILTER((lang(?label) = "en") && (lcase(?label) = lcase("The Godfather")))
    FILTER(YEAR(?release_date) = %d)
    """
    """ % (title.lower(), year)

    response = requests.get(url, params={'query': query})

    # Verifica si la respuesta fue exitosa
    if response.status_code == 200:
        data = response.json()
        return data
    else:
        print(f"Error: {response.status_code}")
        print(response.text)
```

# RESULTADOS DE CONSULTAS

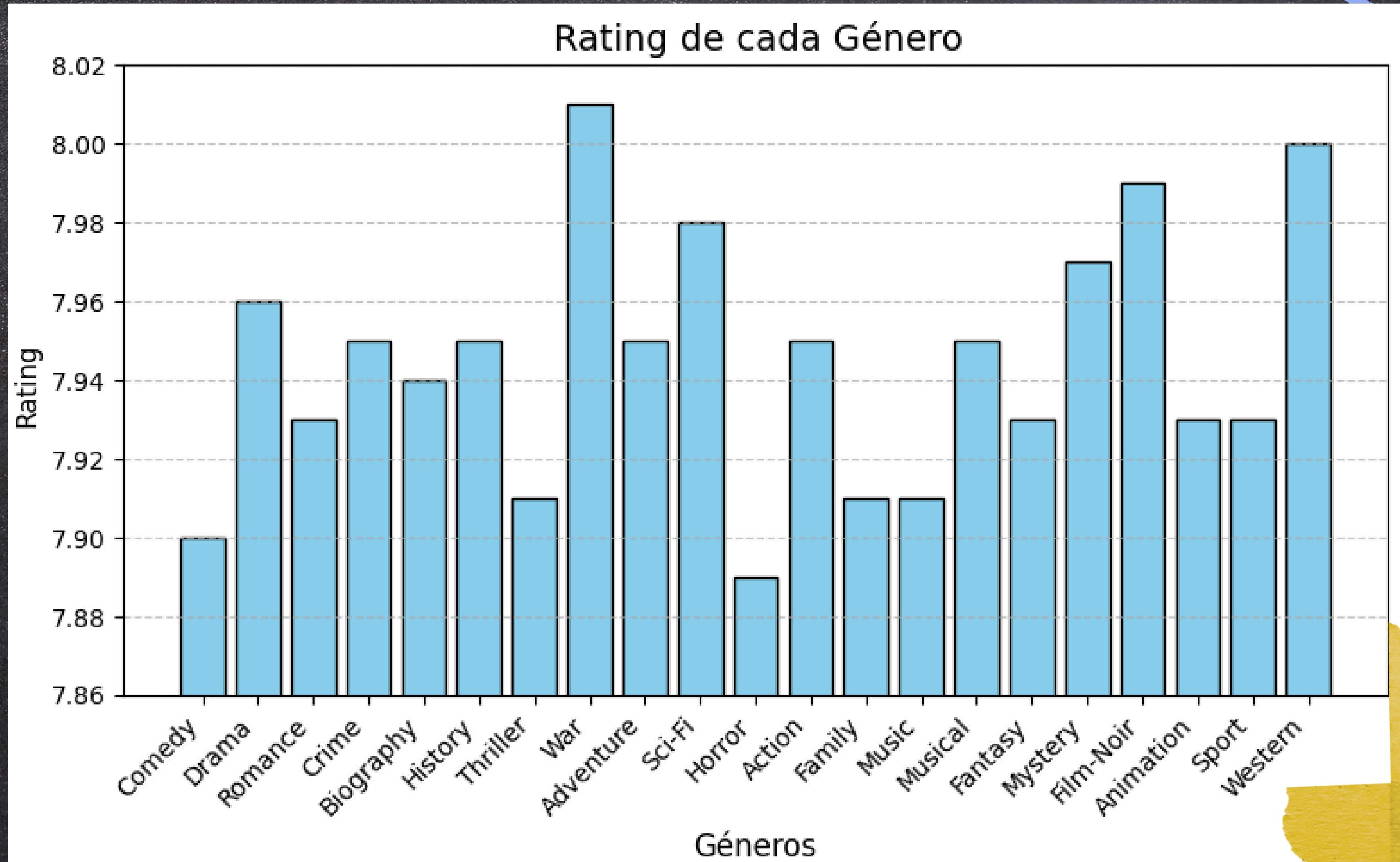


Consulta: encontrar las 5 localizaciones que tienen más películas.

Posición	Localización	Cantidad
1	New York City	96
2	Los Angeles	74
3	London	56
4	Paris	41
5	California	29

# RESULTADOS DE CONSULTAS

Consulta: rating  
por género.



# RESULTADOS DE CONSULTAS



Consulta: el par de actores que aparecen juntos en la mayor cantidad de películas.

Actor 1	Actor 2	Cantidad
Daniel Radcliffe	Rupert Grint	6

# RESULTADOS DE CONSULTAS



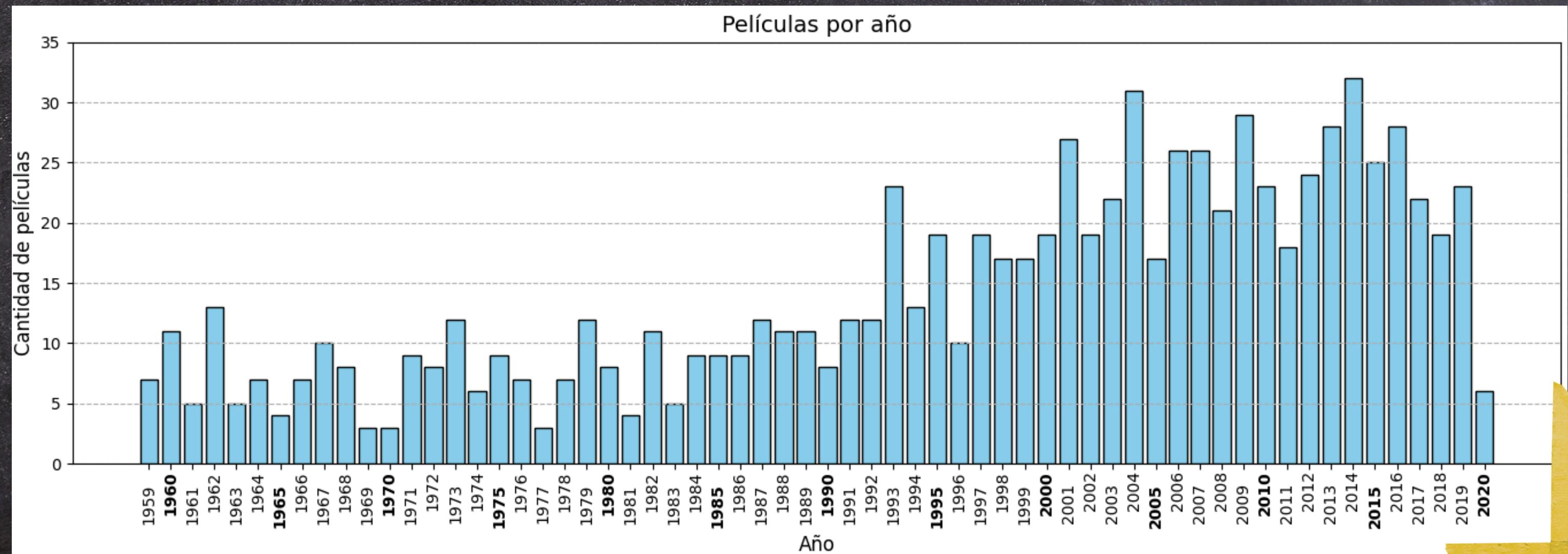
Consulta: top cantidad de películas que tienen el mismo director y actor.

Posición	Director/Actor	Cantidad
1	Charles Chaplin	6
2	Woody Allen	5
3	Clint Eastwood	5
4	Orson Welles	2

# RESULTADOS DE CONSULTAS



Consulta: el conteo de película por año de estreno.



# RESULTADOS DE CONSULTAS



Consulta: top 5 actores que aparecen en más películas.

Posición	Actor	Cantidad
1	Robert De Niro	17
2	Tom Hanks	14
3	Al Pacino	13
4	Clint Eastwood	12
5	Christian Bale	11

# RESULTADOS DE CONSULTAS



Consulta: los 5 directores que tienen más películas.

Posición	Director	Cantidad
1	Alfred Hitchcock	14
2	Steven Spielberg	13
3	Hayao Miyazaki	11
4	Martin Scorsese	10
5	Akira Kurosawa	10

# RESULTADOS DE CONSULTAS



Consulta: el actor y director que más han trabajado juntos.

Actor	Director	Cantidad de Películas
Toshirô Mifune	Akira Kurosawa	7

# RESULTADOS DE CONSULTAS



Consulta: Las 5 productoras con más películas.

Posición	Productora	Cantidad de Películas
1	Warner Bros.	72
2	Paramount Pictures	48
3	20th Century Studios	44
4	Columbia Pictures	42
5	Universal Pictures	40

# REFLEXIÓN

Lo bueno:

- Aplicamos los contenidos del curso.
- Aprendimos como enriquecer los datos.
- Se obtuvo un dataset más completo y versátil.

# REFLEXIÓN

Lo malo:

- El enriquecimiento de datos no escala bien con más datos.
- Los valores de Grossing y Capital Cost quedaron con formatos distintos.

# REFLEXIÓN

Trabajo futuro:

- Hacer más consultas SPARQL.
- Agregar más datos de distintas fuentes.
- Mejorar el proceso de enriquecimiento para que sea más rápido.

## Yiruzz/IMDb-RDF

Este repositorio contiene los archivos que se utilizaron para el proyecto del curso Web de Datos (CC7220-1) del Departamento de...



2  
Contributors

0  
Issues

0  
Stars

0  
Forks



**Yiruzz/IMDb-RDF: Este repositorio contiene los archivos que se utilizaron para el proyecto del curso Web de Dat...**

Este repositorio contiene los archivos que se utilizaron para el proyecto del curso Web de Datos (CC7220-1) del Departamento de Ciencias de la Computación de la Universidad de Chile. - Yiruzz/IMDb-RDF

 GitHub

# ¡MUCHAS GRACIAS!

