

# Proyecto

Proyecto implementado para Bases de Datos 2

Desarrollado por: **Asunción Mariana Sic Sor**

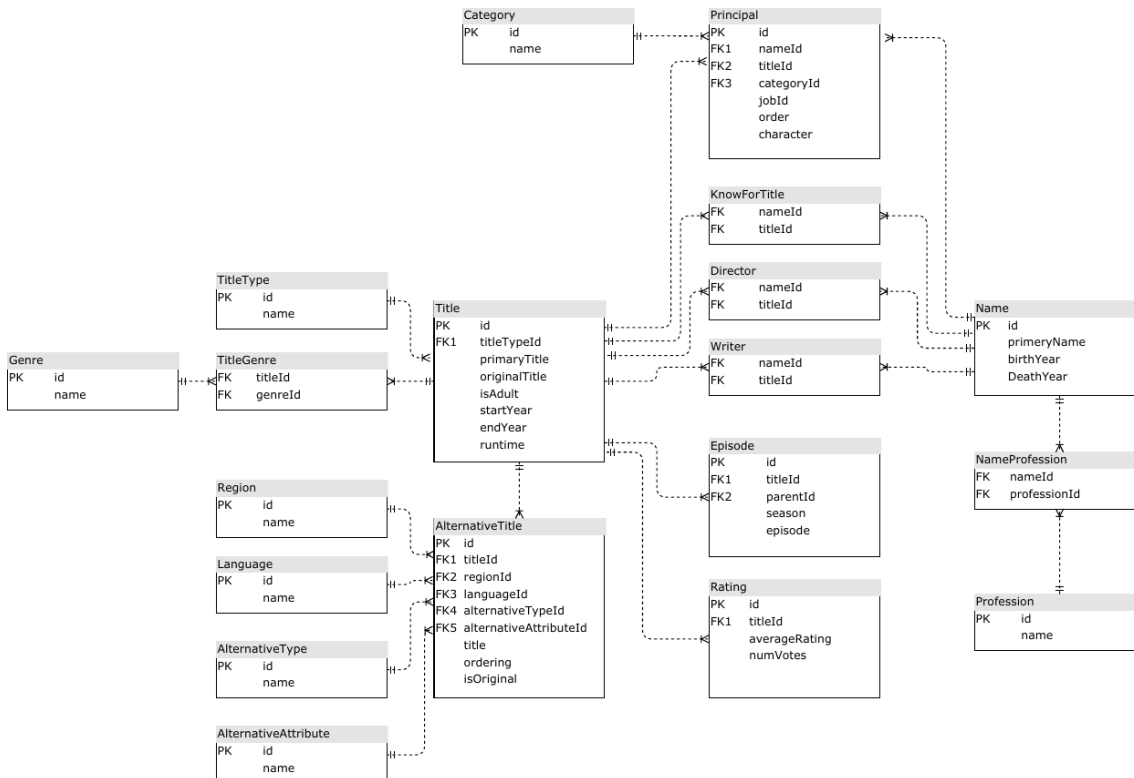
## Tabla de Contenido

- [IMDB](#)
  - [Carga de Datos](#)
  - [Creación de Tablas](#)
- [Netflix](#)
  - [Modelado de Datos](#)
    - [Explicación de Tablas](#)
  - [Inserción a Netflix](#)
    - [Linked Server](#)
    - [Stored Procedure](#)
- [Mongo](#)

## IMDB

Esta base de datos se desarrolló en un servidor con Sistema Operativo Arch Linux (Manjaro KFCE)

El modelo Entidad-Relacional es el siguiente



Esta base de datos se ha desarrollado en SQL Server y los datos se encuentran en este [enlace](#).

## Carga Datos

Para cargar dichos datos, se crea una tabla temporal por cada archivo `.tsv` de la siguiente manera

*Para el procedimiento se utiliza el archivo `namebasics.tsv`*

Primero se crea la tabla temporal para almacenar los datos

```
CREATE TABLE namebasics(  
    nconst VARCHAR(115),  
    primaryname VARCHAR(115),  
    birthyear VARCHAR(45),  
    deathyear VARCHAR(45),  
    primaryProfession VARCHAR(115),  
    knownforTitles VARCHAR(115)  
);
```

Luego se insertan los datos con la siguiente instrucción

```
BULK INSERT namebasics  
FROM '/namebasics.tsv'  
WITH  
(  
    FIRSTROW = 2,  
    FIELDTERMINATOR = '\t',  
    ROWTERMINATOR = '\n'  
);
```

Y los datos se cargaran

```

26
25 -- CREACION DE TABLAS TEMPORALES
24
23
22
21
20
19
18
17
16
15 BULK INSERT namebasics
14 FROM '/namebasics.tsv'
13 WITH
12 (
11     FIRSTROW = 2,
10     FIELDTERMINATOR = '\t',
9     ROWTERMINATOR = '\n'
8 );
7
6
5 CREATE TABLE namebasics(
4     nconst VARCHAR(115),
3     primaryname VARCHAR(115),
2     birthyear VARCHAR(45),
1     deathyear VARCHAR(45),
0     primaryProfession VARCHAR(115),
-1    knownforTitles VARCHAR(115)
-2
-3
-4
-5
-6
-7
-8
-9
-10
-11
-12
-13
-14
-15
-16
-17
-18
-19
-20
-21
-22
-23
-24
-25
-26

```

## Messages

4:15:02 PM      Started executing query at Line 15  
 (10046391 rows affected)  
 Total execution time: 00:02:52.007

Dicho procedimiento se ejecuta para todos los archivos proporcionados

## Creación-Inserción Tablas

Luego se crean las tablas según el [modelo entidad/relación](#) correspondiente.

*Por ejemplo, se muestra con la tabla Category*

Se crea la tabla

```

create table genre(
    id int IDENTITY(1,1),
    name VARCHAR(75),
    PRIMARY KEY (id)
);

```

Luego con una consulta a la base de datos de la tabla temporal `titleprincipals` se inserta a la tabla

```
INSERT into category
SELECT distinct category from titleprincipals;
```

Y ya quedan insertados los datos en la base de datos

home > marianasic > Documents > bases2 > Proy-Bases2 > IMDB > insert.sql

Run Cancel Disconnect Change Connection IMDB Explain Enable SQLCMD

```
1 INSERT into category
2 SELECT distinct category from titleprincipals;
3
4 select * from category;
```

Results Messages

	id	name
1	1	actor
2	2	self
3	3	writer
4	4	editor
5	5	archive_footage
6	6	producer
7	7	actress
8	8	archive_sound
9	9	composer
10	10	director
11	11	cinematographer
12	12	production_designer

De esta manera se insertan los datos en la base de datos.

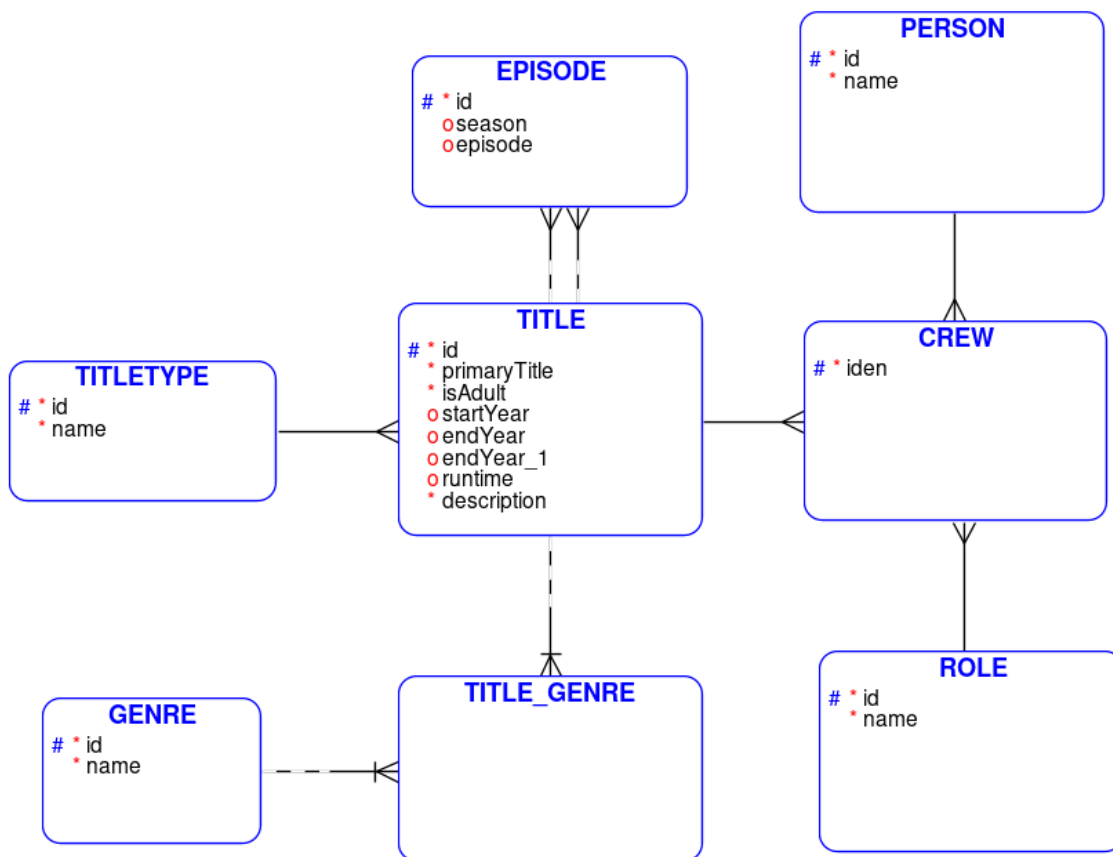
El script de la inserción en la base de datos se encuentra en el archivo [IMDB/insert.sql](#)

# Netflix

Para la base de datos de este sistema, se desarrolló en SQL Server en una máquina virtual con [Xubuntu](#) como Sistema Operativo

## Modelado de Datos

El modelo entidad relación surge a partir del diagrama de [IMDB](#), pero para la base de datos de Netflix se reduce al siguiente diagrama

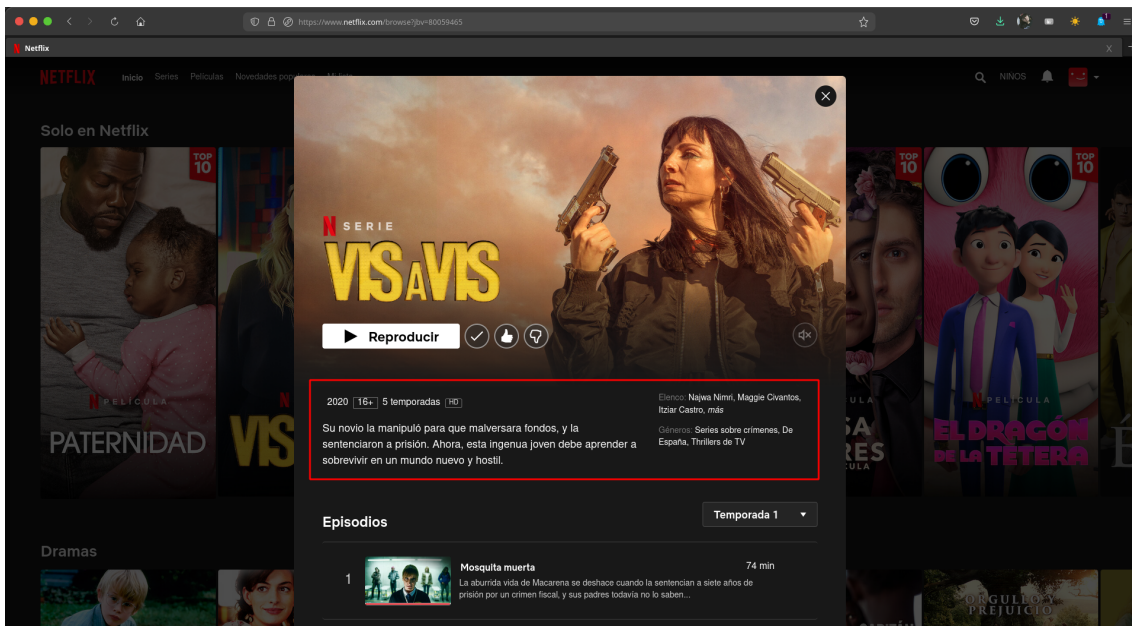


Su modelo relacional se encuentra [aquí](#)

## Tablas

Tabla	Descripción
Genre	En esta tabla se almacenan los géneros posibles de algún título
TitleType	Se almacenan los posibles tipos de un título. Ej. Show, película, serie, etc.
Title	En esta tabla se almacena la información relevante de un título y en ella también hace referencia con llave foránea hacia los posibles tipos de títulos y géneros que pueda clasificarse un mismo título. El valor de esta llave primaria es exactamente igual a la de IMDB para poder vincular su rating
Title_Genre	A esta tabla caen todos los géneros posibles en los que pueda clasificarse un mismo título.
Episode	En dado caso el título sea una serie, acá se encuentra a detalle su único o varios episodios correspondientes a un mismo título
Person	Esta tabla almacena todas las personas que pueden verse enrollada en algún título
Role	Acá se especifican los posibles roles que las personas pueden ejercer. Éstos pueden ser actor, actriz, director o escritor
Crew	Acá se definen todas las personas y rol que ejercen en un mismo título

Se eligió dicho modelo basándose en los datos que actualmente muestra el catálogo de [Netflix](#) ©



**Fuente:** Esta imagen es propiedad de [Netflix ©](#) y en ella se muestra el título de [Vis a Vis](#)

La creación de las tablas descritas se encuentra en [Netflix/create.sql](#)

## Inserción a Netflix

Para estar insertando nuevas películas en la base de datos de Netflix, se necesita un procedimiento almacenado y que la película que se vaya insertando cree un vínculo entre la base de datos IMDB con la de Netflix, dado que ambas están en diferente servidor, se crea un [Linked Server](#) para conectar entre servidores.

### Linked Service

El servicio enlazado se crea con la siguiente instrucción

```
USE [NETFLIX]
GO

EXEC sp_addlinkedserver
    @server = N'192.168.1.2',
    @srvproduct = N'SQL Server';
GO
```

La dirección IP 192.168.1.2 pertenece al servidor que aloja la base de datos de IMDB. Para verificar que se haya enlazado correctamente, se ejecuta la consulta

```
SELECT name FROM [192.168.1.2].master.sys.databases;
```

Para rectificar que entre las bases de datos del otro servidor, se encuentre la de IMDB

```

6 EXEC sp_addlinkedserver
7     @server = N'192.168.1.2',
8     @srvproduct = N'SQL Server';
9 GO
10
11 SELECT name FROM [192.168.1.2].master.sys.databases;

```

## Results Messages

	name
1	master
2	tempdb
3	model
4	msdb
5	IMDB

La inserción de datos simples como lo son las personas, los roles, las categorías y tipos de título se encuentran en el archivo [insert.sql](#) con el servicio enlazado.

## Stored Procedure

Para llenar las tablas `title`, `episode`, `title_genre` y `crew` se hace uso de un procedimiento almacenado

Se comienza creando y asignando el título de `insert_title` y con parámetros:

Parámetro	Descripción
@titulo	Es de tipo varchar, contiene el nombre del título que se desea almacenar
@anio	Es de tipo int, contiene el año de inicio, osea el año en el que se inició o estrenó el título a almacenar
@tipo_titulo	Es de tipo int, contiene el tipo de título que es (ya sea, tvSerie, show, movie, etc)
@desc	Es de tipo varchar, contiene una breve descripción del título que se va a almacenar
@flag	Es de tipo varchar, parámetro de salida para imprimir el resultado del procedimiento almacenado

Se crea el procedimiento y sus parámetros:

```

CREATE PROCEDURE insert_title
    @titulo varchar(600),
    @anio int,

```

```

@tipo_titulo int,
@desc varchar(5000)
@flag varchar(115) OUTPUT

```

Luego se hace uso de una variable `@id_title` la cual almacena el id del título en dado ya esté registrado en la base de datos de IMDB

```

DECLARE @id_title VARCHAR(115);

SET @id_title = (
    SELECT TOP 1 tl.id FROM [192.168.1.2].IMDB.dbo.title tl
    WHERE UPPER(tl.primaryTitle) LIKE UPPER(@titulo) AND tl.startYear = @anio AND
    tl.titleTypeId = @tipo_titulo
);

```

Se hace la respectiva verificación si la variable es nula o no (es decir, si el título existe o no). Si no existe, se sale del procedimiento

```

IF @id_title IS NULL
BEGIN
    -- SIGNIFICA QUE EL TITULO NO ESTA EN EL CATALOGO DE IMDB
    SET @flag = 'El título ' + @titulo + ' no se ha encontrado.';
    RETURN
END

```

En dado caso ya exista, se inserta en las tablas respectivas que se han mencionado antes.

Además se vuelve a verificar que no exista el título a agregar en la base de datos de Netflix

```

IF (SELECT primaryTitle FROM title t WHERE t.id = @id_title) IS NULL
BEGIN
    -- INSERTAR A LA TABLA TITLE
    INSERT INTO title(id, primaryTitle, isAdult, startYear, endYear, runtime,
[description], titleTypeId)
    SELECT TOP 1 tl.id, tl.primaryTitle, tl.isAdult, tl.startYear, tl.endYear,
tl.runtime, @desc, tl.titleTypeId
    FROM [192.168.1.2].IMDB.dbo.title tl WHERE tl.id = @id_title;

    -- INSERTAR A LA TABLA CREW
    INSERT INTO crew(titleId, personId, roleId)
    SELECT TOP 3 pr.titleId, pr.nameId, role.id
    FROM [192.168.1.2].IMDB.dbo.principal pr INNER JOIN
[192.168.1.2].IMDB.dbo.category cg ON pr.categoryId = cg.id
    INNER JOIN role ON role.name = cg.name
    WHERE pr.titleId = @id_title AND (cg.name = 'actor' OR cg.name = 'actress');

    INSERT INTO crew(titleId, personId, roleId)
    SELECT TOP 1 dr.titleId, dr.nameId, role.id
    FROM [192.168.1.2].IMDB.dbo.director dr, role
    WHERE dr.titleId = @id_title AND role.name = 'director';

    INSERT INTO crew(titleId, personId, roleId)
    SELECT TOP 1 dr.titleId, dr.nameId, role.id

```



```

FROM [192.168.1.2].IMDB.dbo.director dr, role
WHERE dr.titleId = @id_title AND role.name = 'writer';

-- INSERT TITLE_GENRE
INSERT INTO title_genre(genreId, titleId) VALUES (@tipo_titulo, @id_title);

-- INSERTAR A LA TABLA EPISODE
IF (SELECT count(*) FROM [192.168.1.2].IMDB.dbo.episode ep WHERE ep.parentId =
@id_title) > 0
    BEGIN
        INSERT INTO episode(titleId, parentId, season, episode)
        SELECT ep.titleId, ep.parentId, ep.season, ep.episode
        FROM [192.168.1.2].IMDB.dbo.episode ep WHERE ep.parentId = @id_title;
    END

SET @flag = 'El título ' + @titulo + ' ha sido agregado.';
RETURN
END

```

Y si ya ha sido agregado antes, no se agrega nada en la base de datos

```

ELSE
    BEGIN
        SET @flag = 'El título ' + @titulo + ' ya ha sido agregado antes.';
        RETURN
    END

```

El procedimiento almacenado completo se encuentra en el archivo [stored.sql](#)

Luego ya solo se ejecuta con la instrucción

```

DECLARE @result VARCHAR(115);
EXEC insert_title 'Frozen',2013,9,'Elsa found her power', @flag = @result OUTPUT;
SELECT @result 'Resultado';

```

```

73 |
74 | DECLARE @result VARCHAR(115);
75 | EXEC insert_title 'Frozen',2013,9,'Elsa found her power', @flag = @result OUTPUT;
76 | SELECT @result 'Resultado';
77 |

```

## Results Messages

	Resultado
1	El título Frozen ha sido agregado.