



OBJETIVO

- Almacenar relación existente entre películas y actrices.
- Se debe leer el HTML del sitio de actrices y almacenar lo que se le indica en una base de datos.
- Proyecto se compone de: Productores y Consumidores,

WEBSCRAPPING: PRODUCTORES Y CONSUMIDORES



GO

Crear y manejar
productores y
consumidores.



Docker - Mysql

Mysql para guardar todos
los datos.

EJECUCIÓN



DOCKER. docler-compose up



GO: ./pcb #Productores #Consumidores



se refiere a la cantidad

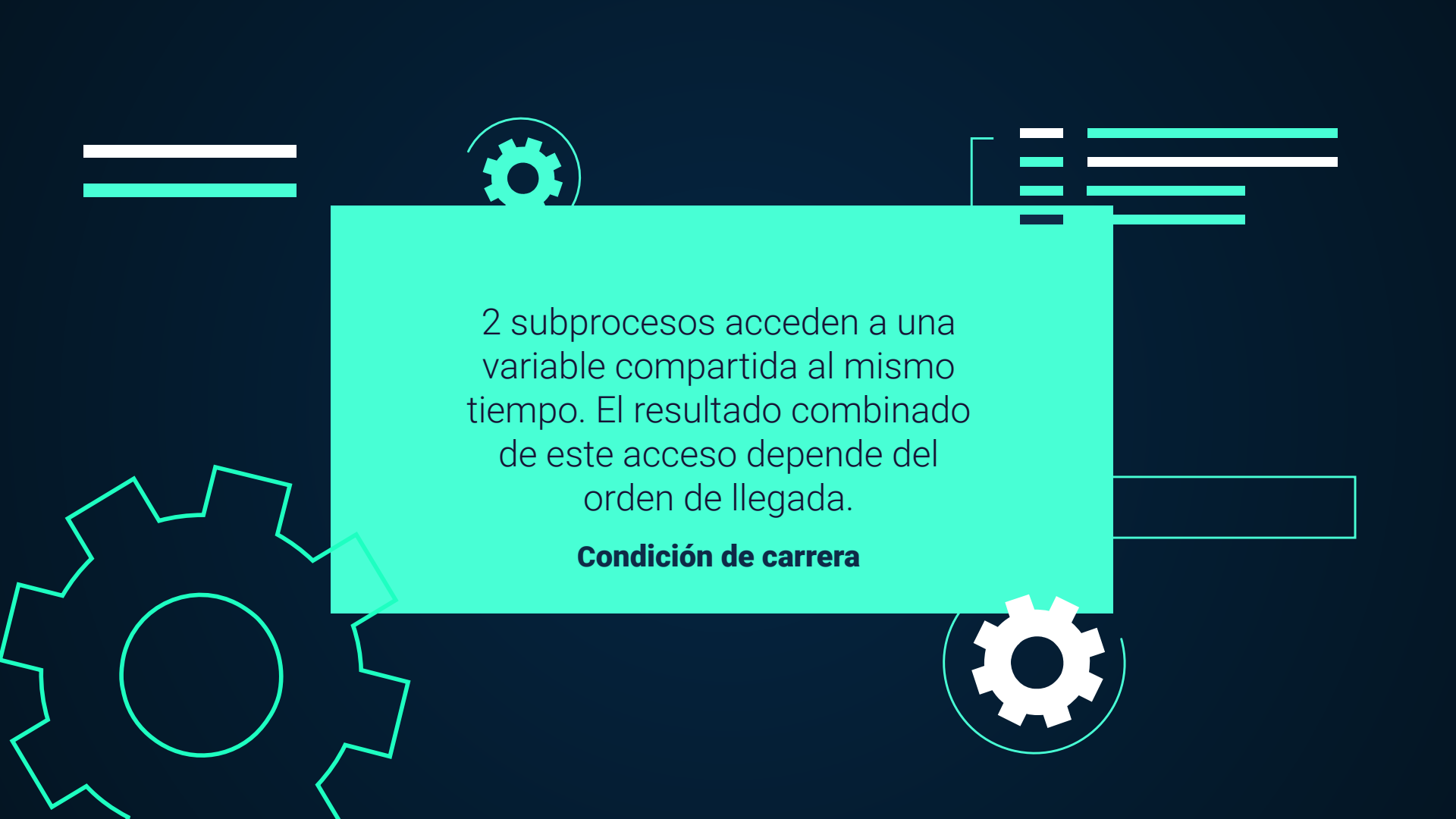




EJECUCIÓN

Crear cantidad de productores que recojan los datos de cada actriz.

-> Esto es pasado a los consumidores sin que ocurra una condición de carrera que haga que los datos se repitan.

The background is a dark blue gradient. In the top left, there are two horizontal bars, one white and one red. In the top center, a small red gear is enclosed in a white circle. In the top right, there are several horizontal bars in white and red. In the bottom left, a large white gear is partially visible. In the bottom right, a small white gear is enclosed in a white circle. A large white rectangle is centered on the slide, containing text.

2 subprocessos acceden a una variable compartida al mismo tiempo. El resultado combinado de este acceso depende del orden de llegada.

Condición de carrera

File Edit Selection View Go Run Terminal Help

EXPLORER

- PCB
 - > docker-compose
 - orm
 - orm.go
 - web_scraping
 - web_scraping.go
 - go.mod
 - go.sum
 - pcb
 - pcb.go

PROBLEMS

TERMINAL

OUTPUT

DEBUG CONSOLE

```
Inicia transaccion de actriz: Amy Davidson
Inicia transaccion de actriz: Minnie Driver
Inicia transaccion de actriz: Terry Farrell
Inicia transaccion de actriz: Idalis DeLeón
Inicia transaccion de actriz: Alice Faye
Inicia transaccion de actriz: Beanie Feldstein
Inicia transaccion de actriz: Grey DeLisle
Inicia transaccion de actriz: Holly Fields
Inicia transaccion de actriz: Vanessa Ferlito
Inicia transaccion de actriz: Jere Fields
Inicia transaccion de actriz: Kim Fields
Inicia transaccion de actriz: Joanne Dru
Inicia transaccion de actriz: Alice Drummond
Inicia transaccion de actriz: Conchata Ferrell
Inicia transaccion de actriz: Ashley Fink
Inicia transaccion de actriz: Ja'Net DuBois
Inicia transaccion de actriz: Jenna Fischer
Inicia transaccion de actriz: Takayo Fischer
Inicia transaccion de actriz: Heather Dubrow
Inicia transaccion de actriz: America Ferrera
Inicia transaccion de actriz: Danielle Fishel
Inicia transaccion de actriz: Haylie Duff
Inicia transaccion de actriz: Carrie Fisher
Inicia transaccion de actriz: Peggy Feury
Inicia transaccion de actriz: Sally Field
Inicia transaccion de actriz: Hilary Duff
Inicia transaccion de actriz: Lori Beth Denberg
Inicia transaccion de actriz: Donna D'Errico
Inicia transaccion de actriz: Chip Fields
Inicia transaccion de actriz: Patti Deutsch
Inicia transaccion de actriz: Julia Duffy
Inicia transaccion de actriz: Noureen DeWulf
Inicia transaccion de actriz: Frances Fisher
Inicia transaccion de actriz: Danielle Fishel
Inicia transaccion de actriz: Fannie Flagg
Inicia transaccion de actriz: Melinda Dillon
Inicia transaccion de actriz: Karen Duffy
Inicia transaccion de actriz: Bridget Fonda
Inicia transaccion de actriz: Olympia Dukakis
Inicia transaccion de actriz: Mia Dillon
Inicia transaccion de actriz: Donna Dixon
Inicia transaccion de actriz: Faith Ford
Inicia transaccion de actriz: Joely Fisher
Inicia transaccion de actriz: Maria Dizzia
Inicia transaccion de actriz: Sutton Foster
Inicia transaccion de actriz: Kaitlin Doubleday
Inicia transaccion de actriz: Sharon Farrell
Inicia transaccion de actriz: Portia de Rossi
Inicia transaccion de actriz: Kate French
Inicia transaccion de actriz: Kirsten Dunst
```

Code

MYSQL WORKBENCH

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' panel shows a tree view with 'sys', 'testapp', 'Tables', 'actresses', 'actress_filmografi', 'films', 'Views', 'Stored Procedures', and 'Functions'. The 'testapp' database is selected. The main editor shows a SQL query with two statements: 'use testapp;' and 'select name, count(name) from actresses group by name having count(name) > 1;'. Below the editor, the 'Result Grid' is empty. The 'Object Info' panel at the bottom left shows the structure of the 'films' table. The 'Result 1' panel at the bottom right shows the execution log with 7 rows of actions and their results.

SCHEMAS

Filter objects

sys

testapp

Tables

actresses

actress_filmografi

films

Views

Stored Procedures

Functions

Limit to 1000 rows

```
1 • use testapp;
2 • select name, count(name) from actresses group by name having count(name) > 1;
```

Result Grid

Filter Rows

Export: Wrap Cell Content

Table: films

Columns:

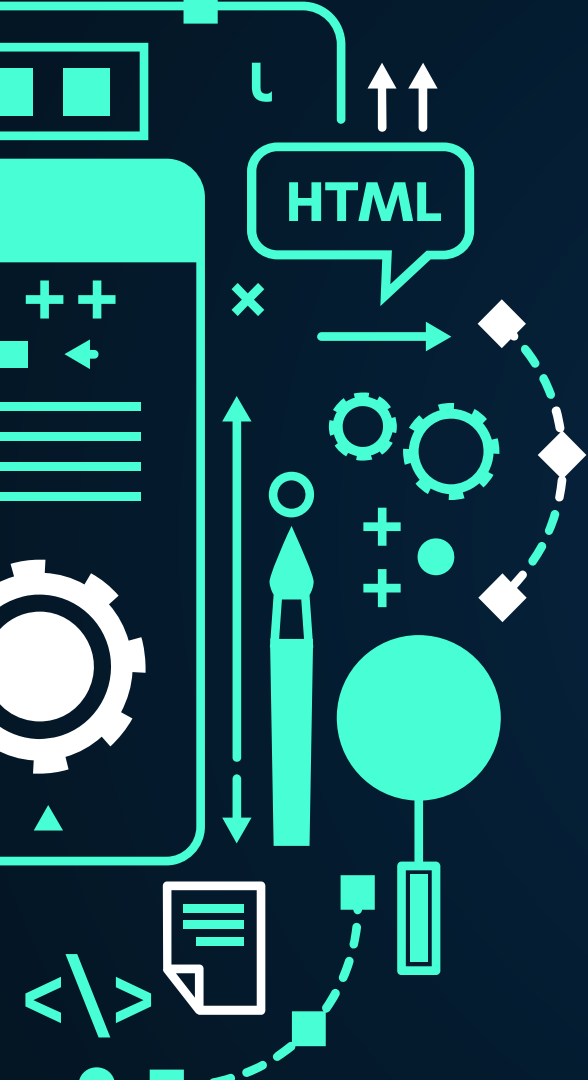
- id_film bigint(20) AI PK
- name longtext
- url longtext
- idconsumidor bigint(20)

Result 1

Action Output

#	Time	Action	Message	Duration / Fetch
1	21:30:15	SELECT * FROM testapp.actresses LIMIT 0, 1000	68 row(s) returned	0.00065 sec / 0.000...
2	00:14:34	SELECT * FROM testapp.actresses LIMIT 0, 1000	1000 row(s) returned	0.0014 sec / 0.0014...
3	00:14:36	SELECT * FROM testapp.actress_filmografias LIMIT 0, 1000	1000 row(s) returned	0.0016 sec / 0.0001...
4	00:14:38	SELECT * FROM testapp.films LIMIT 0, 1000	1000 row(s) returned	0.00077 sec / 0.000...
5	00:15:24	select name, count(name) from actresses group by name...	Error Code: 1046. No database selected Select the default DB to be used by double-clickin...	0.0013 sec
6	00:15:37	use testapp	0 row(s) affected	0.00062 sec
7	00:15:40	select name, count(name) from actresses group by name...	0 row(s) returned	0.011 sec / 0.00001...

Se puede observar que los resultados no muestran ninguna actriz repetida.



MUCHAS GRACIAS