Projecte Plataforma Streaming

Samir M.

David B.

1r DAM

Índex

[Projecte Plataforma Streaming 1](#_Toc135403370)

[Introducció 3](#_Toc135403371)

[M1 4](#_Toc135403372)

[Gestió del Projecte 4](#_Toc135403373)

[Enginyeria del Software 4](#_Toc135403374)

[Codi amb JavaDoc 6](#_Toc135403375)

[Control de versions GIT 7](#_Toc135403376)

[Verificació i validació 8](#_Toc135403377)

[Disseny i Implementació 9](#_Toc135403378)

[M2 11](#_Toc135403379)

[BASE DE DADES 11](#_Toc135403380)

[Model Entitat-Relació 11](#_Toc135403381)

[Model Relacional 12](#_Toc135403382)

[Creació de la BD i Taules 13](#_Toc135403383)

[Inserció Banc de Dades 17](#_Toc135403384)

[Generació Consultes 18](#_Toc135403385)

[M4 19](#_Toc135403386)

[Pàgina web 19](#_Toc135403387)

[Estructura 19](#_Toc135403388)

[Registre i Login 19](#_Toc135403389)

Introducció

En aquest projecte ens han proposat crear una plataforma de Streaming amb una gran varietat de funcionalitats que anirem mostrant al llarg de la memòria, l’aplicació l’hem creat des del llenguatge **Java** amb la **IDE** de **Netbeans**, a més a més hem creat una pàgina per la nostra plataforma Streaming on l’usuari te accés a bastantes funcions, com ara per exemple:

* Registrar-se
* Iniciar sessió
* Modificar dades
* Manipular al seu gust la llista de preferits
* Seleccionar quin tipus de modalitat interessa al usuari i entre altres.

Per a crear la pàgina hem precisat de llenguatge **JavaScript**, **PHP**.

I de llenguatge de marques **HTML** i li hem donat estil amb **CSS**.

Per a guardar totes les dades hem utilitzat **HeidiSQL**(**MariaDB**).

M1

## Gestió del Projecte

### Enginyeria del Software

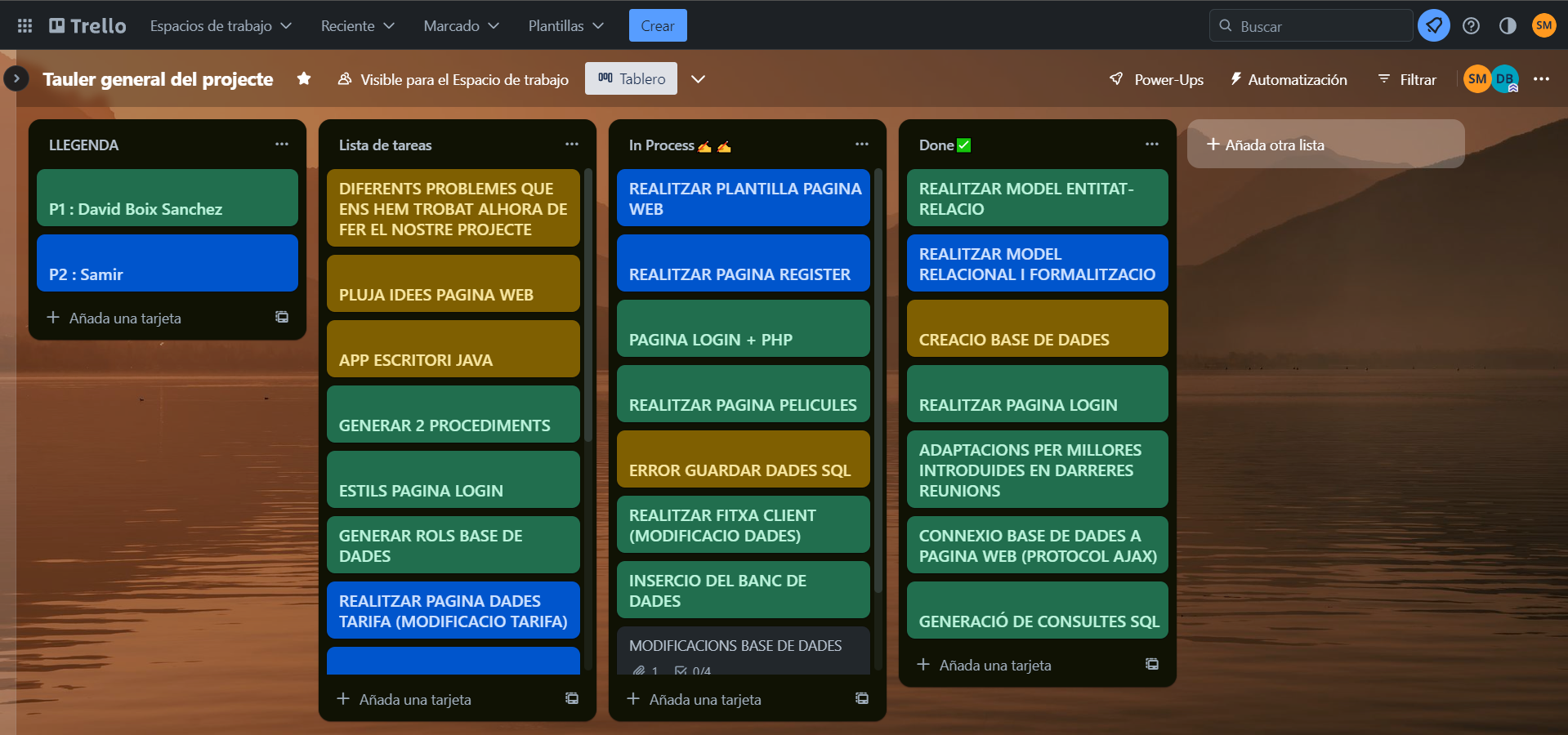
Per a gestionar el nostre projecte i estar al dia hem utilitzat el mètode **Scrum** específicament el **Trello.**

El **Trello** és un software de administració de projectes amb interfície web disponible tant per a mòbil com per a ordinadors.

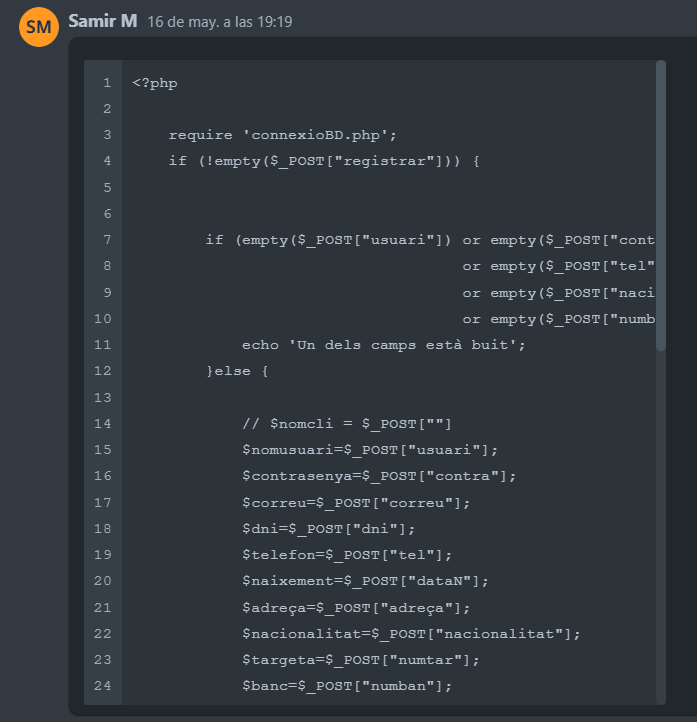


En aquesta imatge es veu com tenim un espai compartit per a poder gestionar el projecte col·lectivament, en la primera secció de targetes, tenim una llegenda on cada membre te un color per a poder dividir la feina i organitzar-nos, ens hem distribuït la feina per colors on cada membre tindrà la tasca del seu color, es a dir si en “*Pere*” te el color vermell haurà de fer les tasques amb la targeta de color vermell.

Seguidament tenim una altra secció de targetes que serien el llistat de targetes de tasques pendents per a fer, l’altra secció serà la feina que està fent cada membre del grup i un cop finalitzada la desplaçarem a la secció de “Done (fet)”.

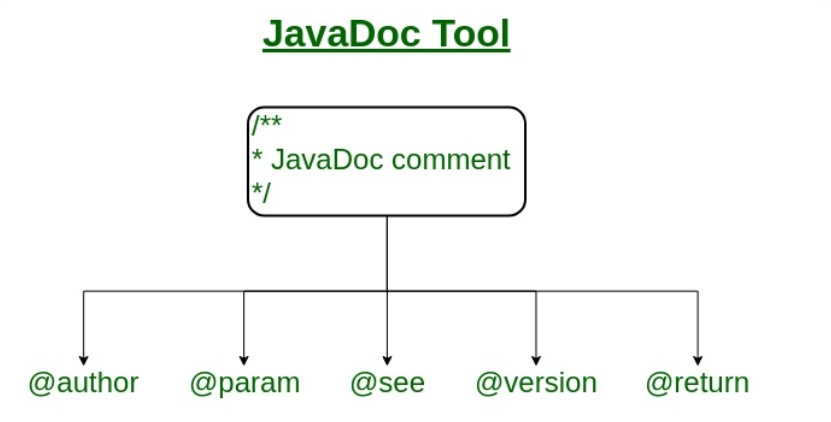


És una eina bastant bona ja que permet adjuntar arxius, quan pengem part del codi ho detecta i ho carrega com si fos un arxiu.



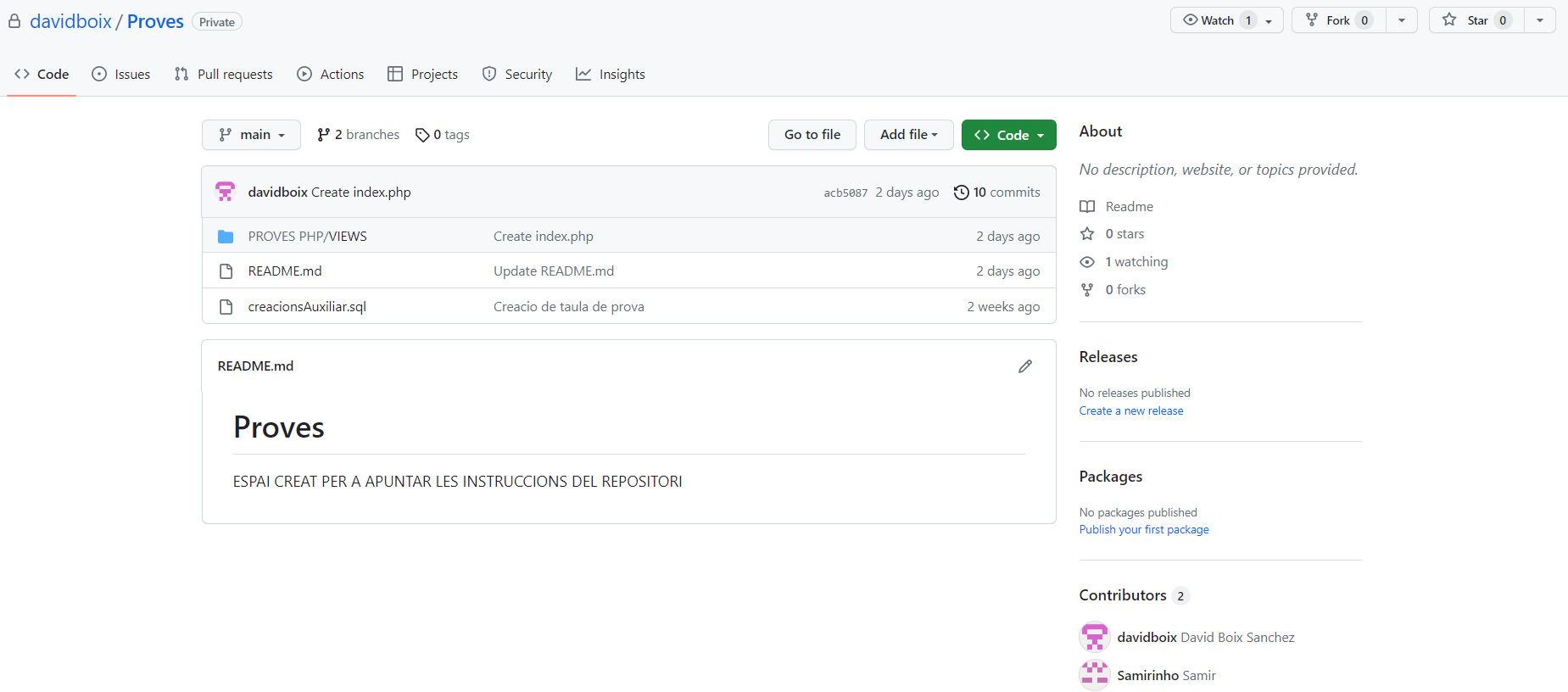
### Codi amb JavaDoc

JavaDoc es una eina molt útil que ens ofereix Oracle però també la podem utilitzar a la nostra IDE de **Netbeans** serveix per a documentar el codi del nostre programa o aplicació, com hem dit anteriorment es una eina molt important ja que ens ajudarà per fer manteniment del codi o per a saber que fa cada mètode o funció.



### Control de versions GIT

Utilitzarem el GitHub per a emmagatzemar i poder manipular l’aplicació, es una eina que ens ajudarà molt si participem en un projecte amb grup ja que permet col·laboradors i entre els dos podeu modificar i emmagatzemar el codi, també ens va be per a fer commits del nostre codi que això dona més seguretat a l’aplicació ja que guardem una instantània del programa en un cert punt.

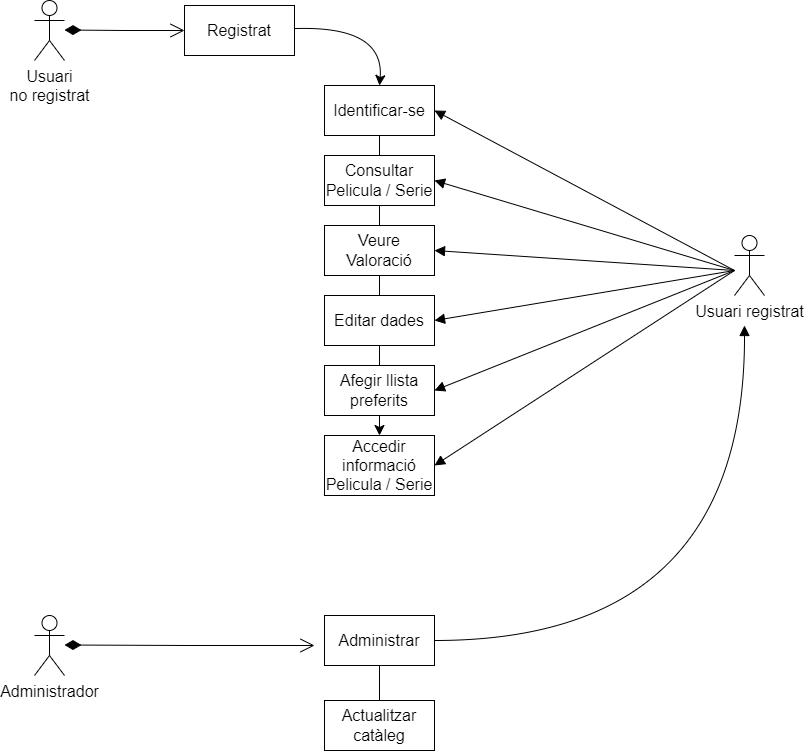


Aquest es un exemple de repositori creat per un dels membres, com podeu veure a la part inferior-dreta es veuen els contribuïdors del repositori. Els fitxers/carpetes creades, els **commits** fets, un ReadMe que pot anar be per entendre l’aplicació.

### Verificació i validació

### Disseny i Implementació

El disseny de diagrama **estàtic** que vam pensar per a la nostra aplicació va ser el següent:



La principal diferencia entre els dos tipus de diagrames és que el **diagrama de classe** el que fa es descriure la estructura estàtica del sistema i el **diagrama de casos** **d’us** es una descripció d’accions o funcions que te un sistema i hi participen actors.

El disseny de diagrama de **classes** es el següent:

[IMATGE DEL DIAGRAMA DE CLASSES]

Per a fer-lo vam utilitzar un programari web per a diagrames anomenat [diagrams](https://app.diagrams.net/?src=about).

# M2

## BASE DE DADES

### Model Entitat-Relació

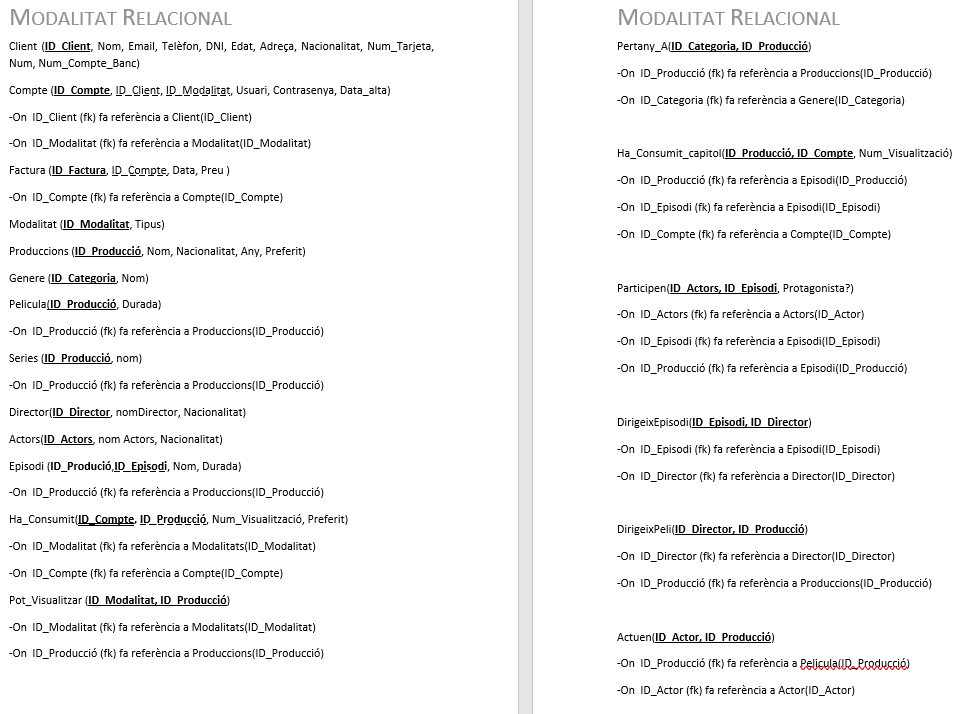
El model **Entitat** - **Relaci**ó ens permet representar de manera simplificada els components que ens en un procés de negoci i la manera com aquests es relacionen entre si.

Captura de com vam fer el nostre model **entitat** - **relació** sobre el projecte.

### Model Relacional

El model relacional ens permet construir estructures de dades per representar les diferents informacions del món real que tinguin algun interès. Les estructures de dades construïdes seguint el model relacional estan formades per conjunts de relacions.

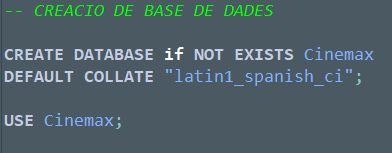
Captura de com vam fer el nostre model relacional sobre el projecte.



Seguint el model entitat-relació vam aconseguir fer-lo.

### Creació de la BD i Taules

En primer lloc mostrarem com hem creat la Base.

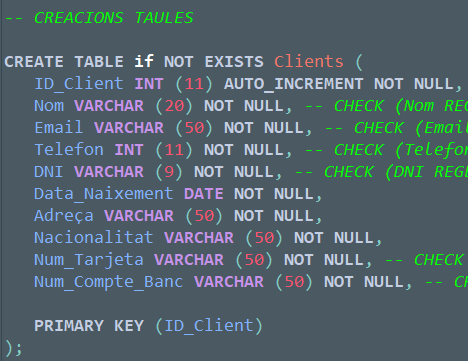


Amb el **if not exists** el que fem es crear la base de dades només si no existeix.

I amb el **default collate** es dir-li a la base de dades amb la llengua i distribució de teclat amb la que treballarem.

I per últim utilitzem el **USE Cinemax** li diem que utilitzi la base de dades creada.

Per crear taules fem el següent:



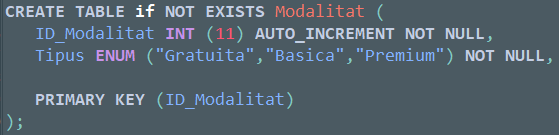
Aquesta seria la estructura per a crear i introduir camps amb els tipus de dades corresponents.

En primer lloc introduïm el nom que volem que tinguin els camps, a continuació marcar de color morat es el tipus de variable que li assignem, les xifres dintre dels parèntesis el valor que li donem a la variable, ho podem deixar buit per deixar el nombre en per defecte.

Fins aquí seria la estructura simple per a crear una taula, el que hem afegit nosaltres al camp ID\_Client es un AUTOINCREMENT que la seva funció es fer que vagi augmentant automàticament conforme es vagin creant files de dades als camps. El últim paràmetre que hem utilitzar es el NOT NULL que obliga a no deixar-ho buit.

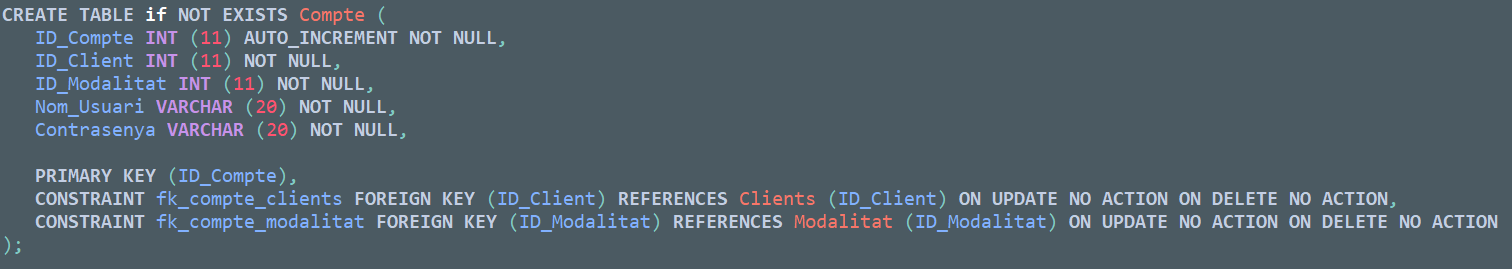
Finalment PRIMARY KEY per a definir la clau principal de la taula.

Creació taula **Modalitat**:

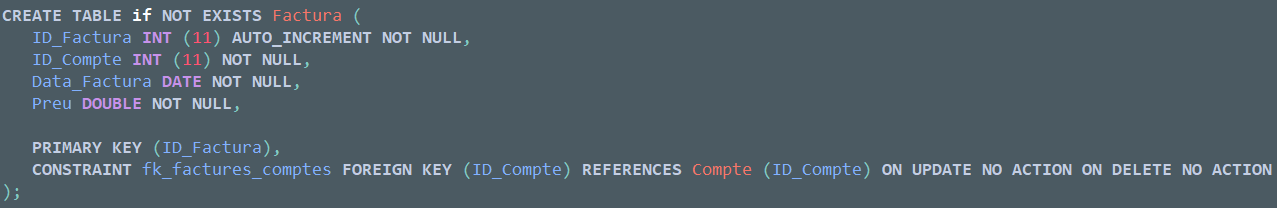


Creació taula **Compte:**

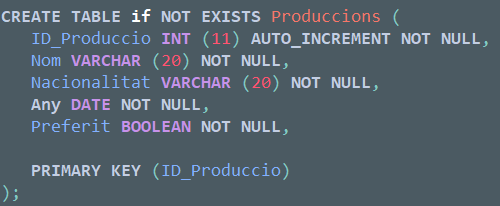
Aquí veiem que hem utilitzat **CONSTRAINTS**, coneixem 2 tipus de constraints, les que utilitzem per crear una restricció en un sol camp i per altra banda la que pot crear restriccions en més de un camp



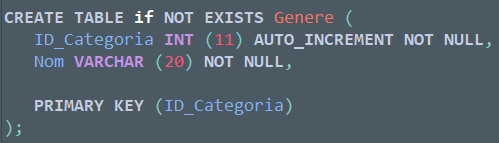
Creació taula **Factura**:



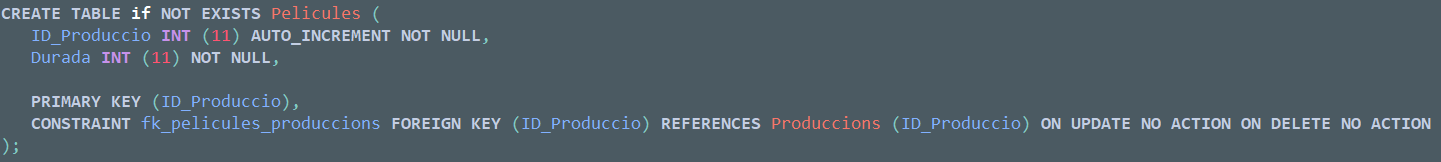
Creació taula **Produccions**:



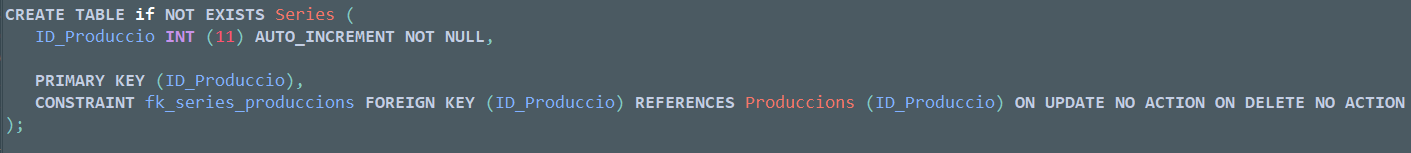
Creació taula **Gènere:**



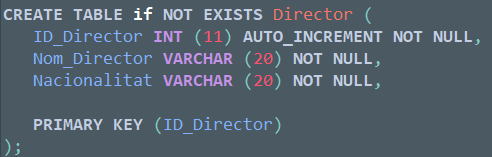
Creació taula **Pel·lícules:**



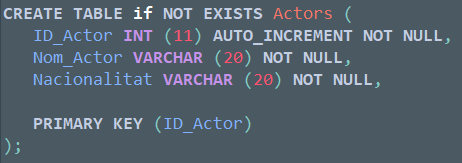
Creació taula **Series**:



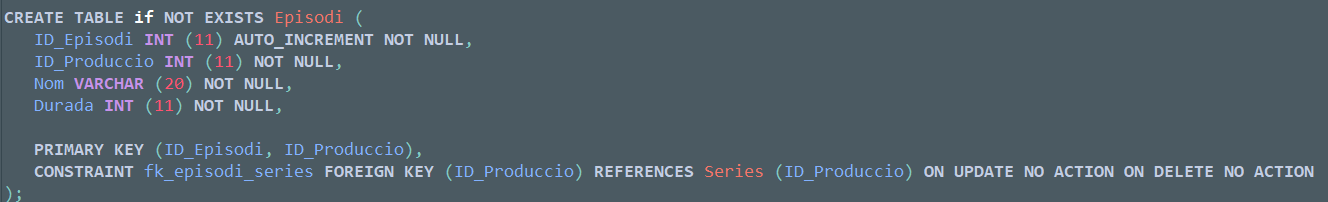
Creació taula **Director:**



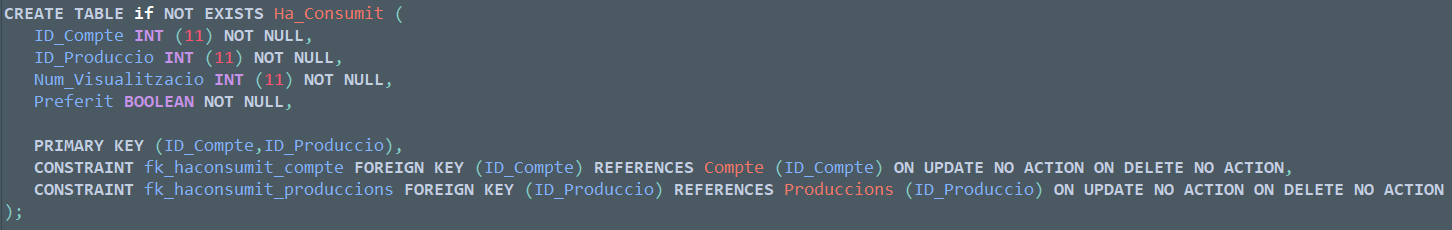
Creació taula **Actors:**



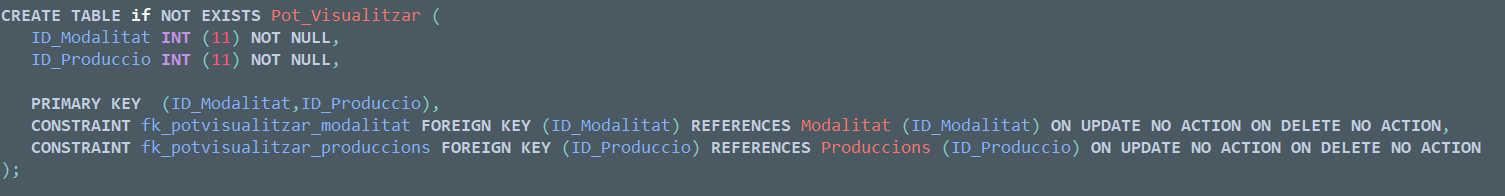
Creació taula **Episodi**:



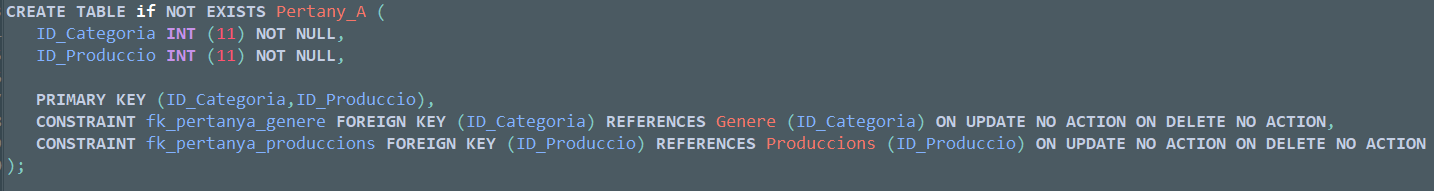
Creació taula **Ha\_consumit:**



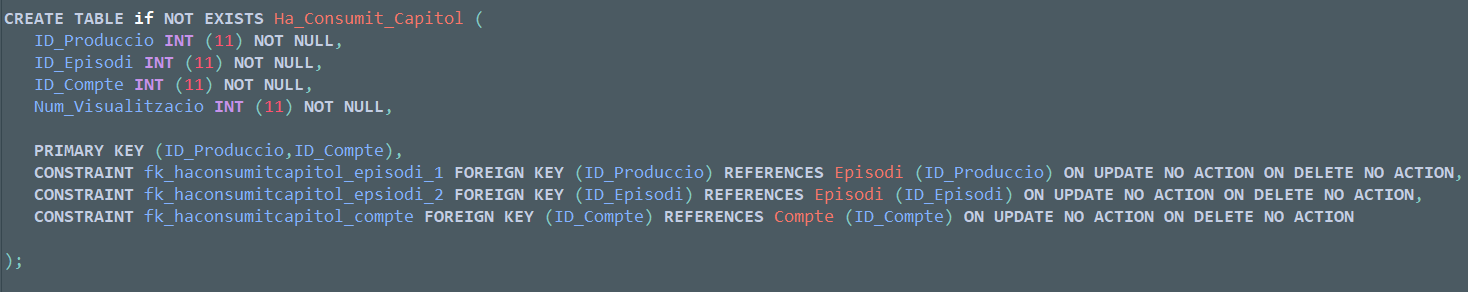
Creació taula **Pot\_Visualitzar**:



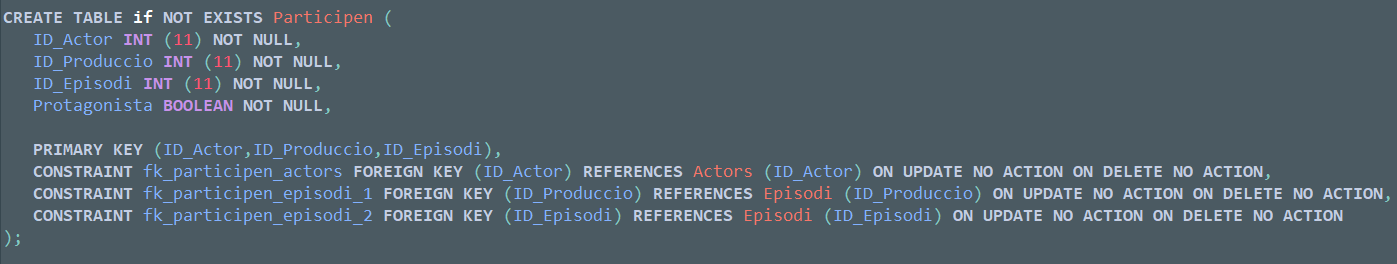
Creació taula **Pertany\_a**:



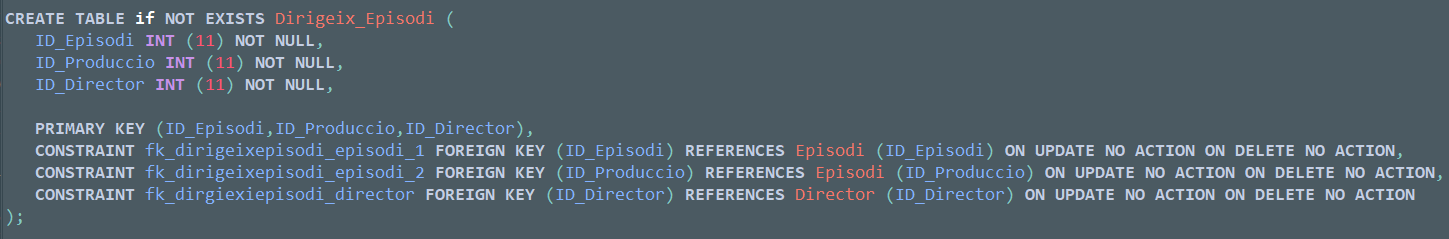
Creació taula **Ha\_Consumit\_Capitol**:



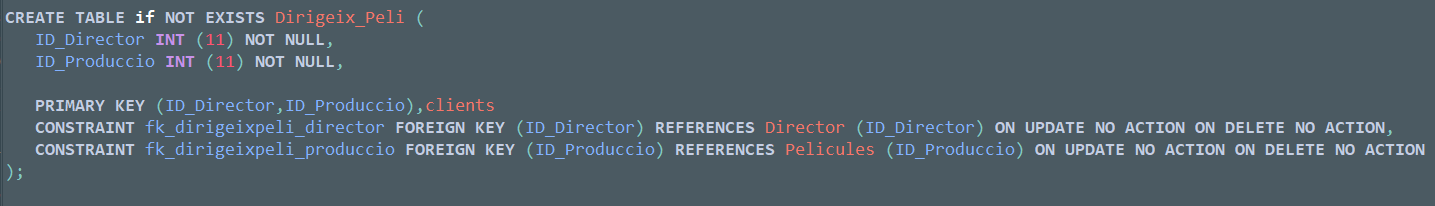
Creació taula **Participen**:

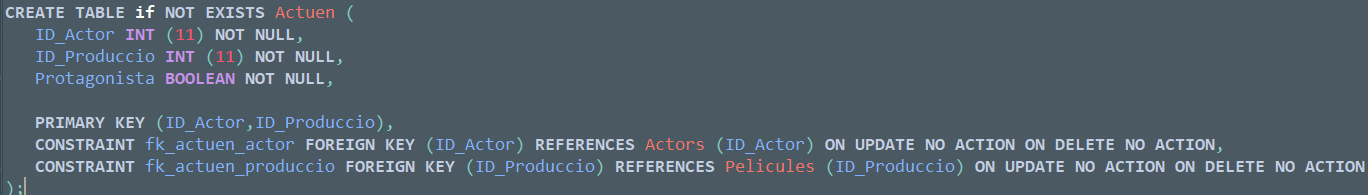


Creació taula **Dirigeix\_episodi**:



Creació taula **Dirigeix\_Peli**:



Creació taula **Actuen**: 

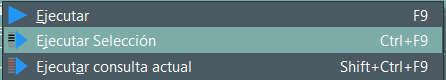
### Inserció Banc de Dades

En primer lloc per a poder inserir el banc de dades hem de comprovar el nom de taules i camps per a no perdre informació ja que podria no introduir-se completament.

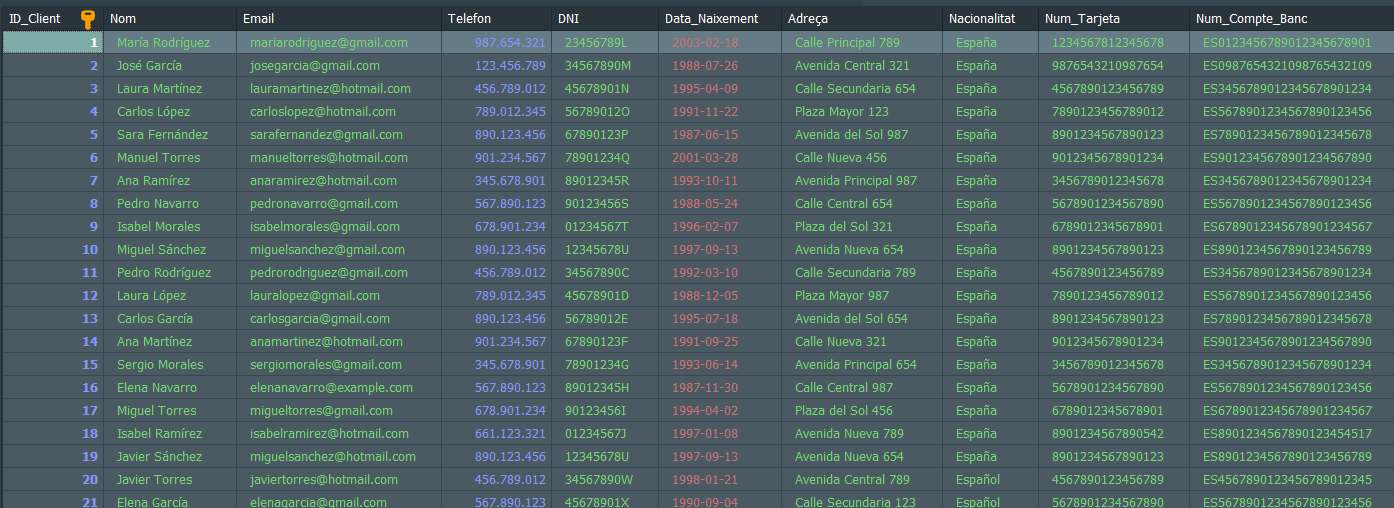
Un cop tinguem la consulta amb el banc de dades i comprovat que tot quadri seleccionarem el codi i començarem a executar-lo i comprovar que encaixen les dades.



Aquestes serien les dades que volem introduir al camp **clients** seleccionant el contingut que està dintre de la inserció afegiríem les dades a la taula i als seus camps corresponents.



Amb un clic dret a la selecció o fent una combinació de tecles **CTRL+F9.**



El resultat hauria de ser semblant, les dades s’han introduït satisfactòriament, ara seria fer el mateix amb totes les insercions restants

### Generació Consultes

Fer les següents consultes

**1º part**

Quines pel·lícules han guardat a la llista (preferits) els clients:

**[Captura]**

Quines pel·lícules / series conformen el top 10  -són les més vistes:

**[Captura]**

Quines pel·lícules / series son el top 10 de les menys vistes:

**[Captura]**

Quins mesos han estat els de més altes de clients:

**[Captura]**

Quin material ha estat donat d’alta en un període determinat de temps:

**[Captura]**

**2º part**

-Genera el rol de directiu, de operador i el rol de client. Atorga els permisos adients sobre les diferents taules de la BD amb aquests rols.

**[Captura]**

-Crea un usuari directorContinguts, administratiu i accesClient i associa’ls als rols respectius.

**[Captura]**

-Genera dos procediments. El primer ha de generar un login automàtic basat en el nom, cognom, procurant que no sigui repetit en la BD. El segon ha de generar un document que indiqui les dades d’un client i el cost total en un periode anual.

**[Captura]**

-Genera un trigger que faci l’actualització del camp de disponibilitat en el moment que es facin noves insercions de series o pel·lícules

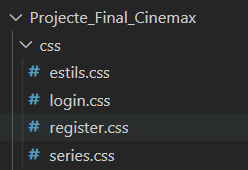
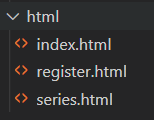
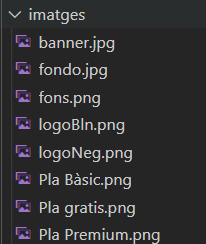
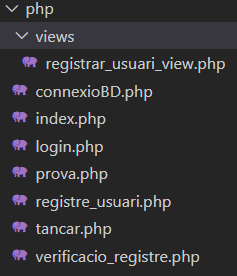
**[Captura]**

# M4

## Pàgina web

### Estructura

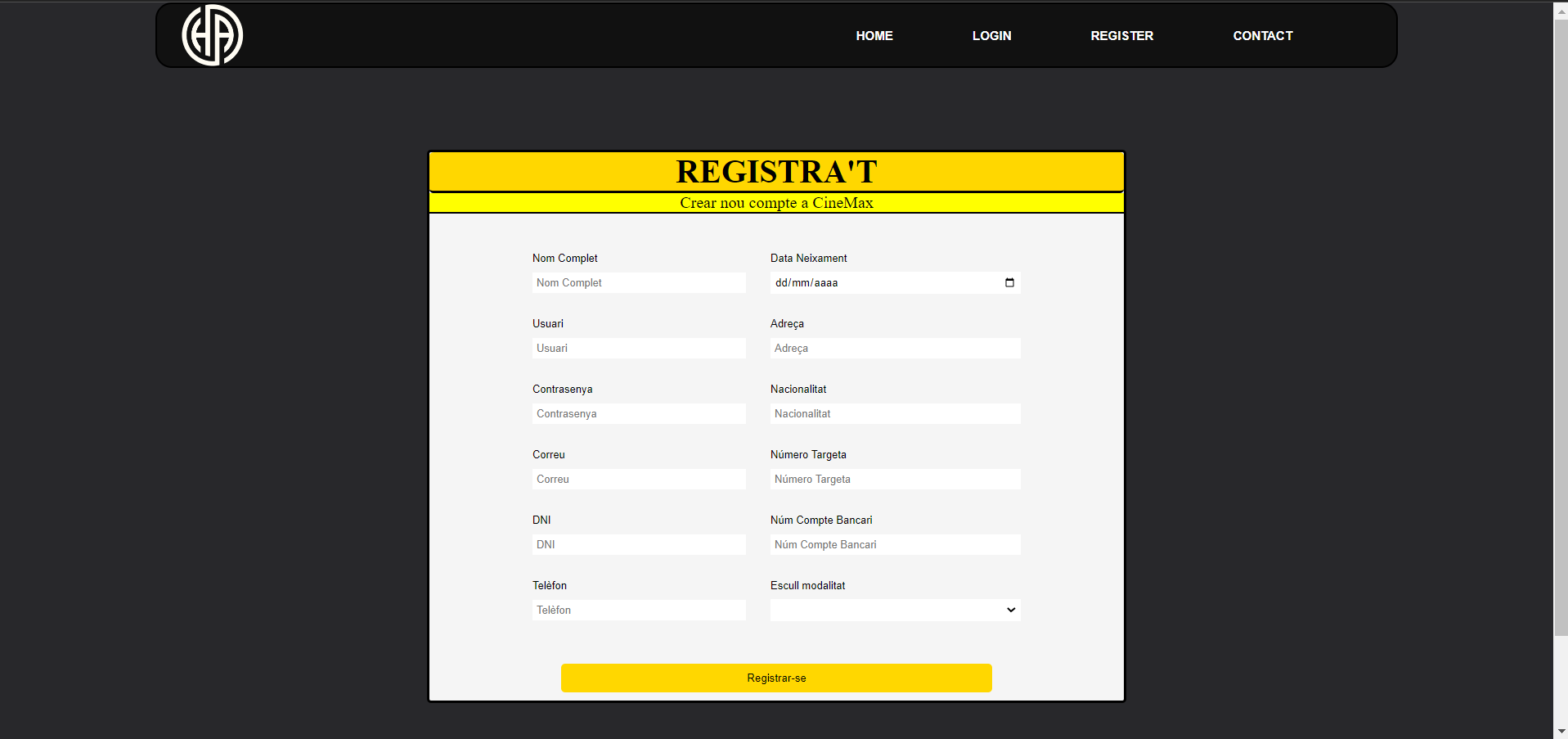
Aquets seria l’estructura de la pàgina web, està organitzat per 5 carpetes on emmagatzemem els fitxers o arxius utilitzats. A simple vista veiem la carpeta **CSS**, **HTML**, **Imatges**, **JS** i **PHP** i una subcarpeta **Views** on tindrem les vistes del **PHP.**

### Registre i Login

**Register**

Aquesta seria la part visual del nostre registre. Disposa de un **Header**, un **formulari** i un **footer**



**Header**

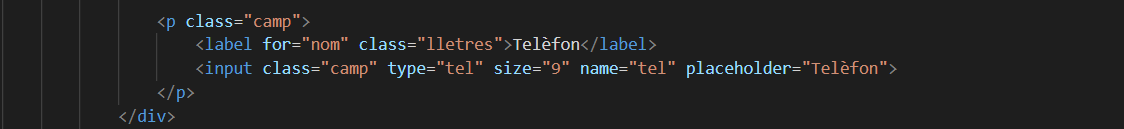


Dintre del **Header** tenim una icona que ens redirigeix al índex, un llistat on tenim accessos redirigits que ens portaran on seleccionem.

**Main**

Dintre del **main** tenim la part més llarga de la pàgina, es on tenim allotjat el formulari del registre, tenim un **div** general que es on està el registre, el formulari el tenim separat per 2 blocs, 2 **div’s** (**bloc1**, **bloc2**) el que fem es separar els inputs per a crear 2 caixes i distribuir-les millor dintre del formulari.

Bloc1

Bloc 2



Amb el paràmetre **name** el que fem es guardar les dades introduïdes al input.

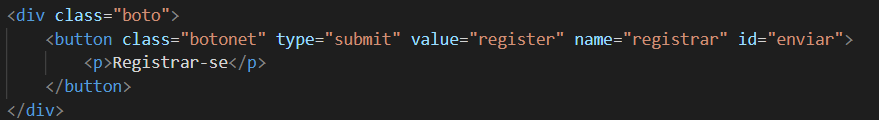
Amb el paràmetre **class** el que fem es donar un nom al contingut per poder donar estil.

Amb el paràmetre **type** el que fem es dir quin tipus serà el camp.

Amb el paràmetre **placeholder** el que fem es posar informació al input.

En el camp de Escull modalitat el que fem es crear un llistat on el usuari podrà seleccionar la modalitat desitjada. I amb el **value** li donem el valor per a poder manipular aquesta dada des de **PHP** i MYSQL.

Finalment el botó per a enviar les dades introduïdes per el client.



**Footer**

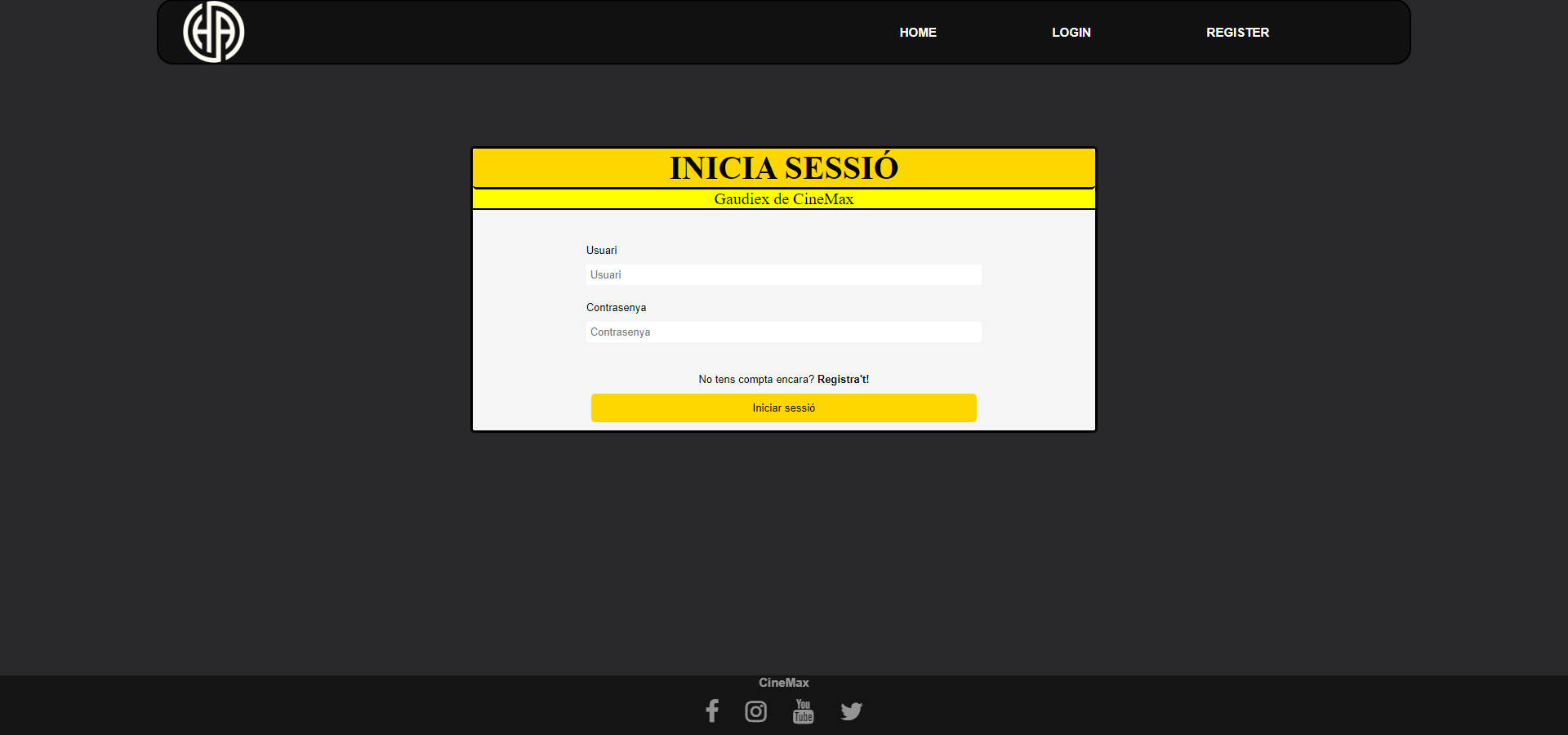
Hem fet un **Footer** bastant senzill i simple, el que conté es el nom de la nostra **Plataforma** **Streaming** i promocionant les xarxes socials que li crearíem, les icones son redirigibles a les pagines pertinents.



Login

El **Login** es similar al **Register**, el únic canvi que sofreix es que té menys camps i en comptes de enviar dades comprova i valida les dades introduïdes.

Part visual del **Login**



**Header**

És idèntic al registre

****

**Main**

Aquest seria el codi HTML per a iniciar sessió.

Amb el **action** el que fem es indicar-li on enviar les dades del formulari.

Amb el **method** li estem indicant al formulari es com transferir les variables del formulari

(get o post).

****

**Footer**



El **Footer** es idèntic al del registre.