

## Proyecto Final Taller de Aplicaciones

Generado por Doxygen 1.9.0



<b>1 Índice de clases</b>	<b>1</b>
1.1 Lista de clases . . . . .	1
<b>2 Índice de archivos</b>	<b>3</b>
2.1 Lista de archivos . . . . .	3
<b>3 Documentación de las clases</b>	<b>5</b>
3.1 Referencia de la Estructura matriz_struct . . . . .	5
3.1.1 Documentación de los datos miembro . . . . .	5
3.1.1.1 columnas . . . . .	5
3.1.1.2 Datos . . . . .	5
3.1.1.3 filas . . . . .	5
3.2 Referencia de la Estructura num_lineas . . . . .	6
3.2.1 Documentación de los datos miembro . . . . .	6
3.2.1.1 columnas . . . . .	6
3.2.1.2 filas . . . . .	6
<b>4 Documentación de archivos</b>	<b>7</b>
4.1 Referencia del Archivo DataShell.h . . . . .	7
4.1.1 Documentación de los 'defines' . . . . .	8
4.1.1.1 EXTERN . . . . .	8
4.1.2 Documentación de las funciones . . . . .	8
4.1.2.1 controlador_linea() . . . . .	8
4.1.2.2 controlador_ProcesoDS() . . . . .	8
4.1.2.3 modelo_Abre_Archivo() . . . . .	9
4.1.2.4 modelo_Columnas() . . . . .	9
4.1.2.5 modelo_Correccion_Nombre() . . . . .	9
4.1.2.6 modelo_Tokenizer() . . . . .	10
4.1.2.7 modelo_Valida_Nombre() . . . . .	10
4.1.2.8 vista_Error_Apertura_Archivo() . . . . .	11
4.1.2.9 vista_Error_Menos1() . . . . .	11
4.1.2.10 vista_MenuDS() . . . . .	11
4.2 Referencia del Archivo Error.h . . . . .	12
4.2.1 Documentación de los 'defines' . . . . .	12
4.2.1.1 EXTERN . . . . .	12
4.2.2 Documentación de las funciones . . . . .	12
4.2.2.1 controlador_errores() . . . . .	12
4.2.2.2 controlador_errorpromedio() . . . . .	13
4.3 Referencia del Archivo Optimizacion.h . . . . .	13
4.3.1 Documentación de los 'defines' . . . . .	13
4.3.1.1 EXTERN . . . . .	14
4.3.2 Documentación de las funciones . . . . .	14
4.3.2.1 controlador_transpuesta() . . . . .	14

4.3.2.2 modelo_optimizacion_content()	14
4.3.2.3 modelo_optimizacion_user()	15
4.4 Referencia del Archivo Peliculas.h	15
4.4.1 Documentación de los 'defines'	17
4.4.1.1 EXTERN	17
4.4.2 Documentación de los 'typedefs'	17
4.4.2.1 MATRIZ	17
4.4.2.2 TAMANO	17
4.4.3 Documentación de las funciones	17
4.4.3.1 controlador_Proceso()	17
4.4.3.2 DataShell()	18
4.4.3.3 modelo_AbreArchivo()	18
4.4.3.4 modelo_CalculaPromedio()	19
4.4.3.5 modelo_CreaUserDB()	19
4.4.3.6 modelo_DespliegaPeliculas()	19
4.4.3.7 modelo_DespliegaUsuarios()	19
4.4.3.8 modelo_DeterminaTamano()	19
4.4.3.9 modelo_DotRecomendacion()	20
4.4.3.10 modelo_DotSimilares()	20
4.4.3.11 modelo_DotSugerencia()	20
4.4.3.12 modelo_Error()	20
4.4.3.13 modelo_GraficaError()	21
4.4.3.14 modelo_ImprimeRanking()	21
4.4.3.15 modelo_ImprimeUserDB()	21
4.4.3.16 modelo_IniciaUserDB()	22
4.4.3.17 modelo_ObtienLinea()	22
4.4.3.18 modelo_Prediccion()	22
4.4.3.19 modelo_RecorreTXT()	23
4.4.3.20 modelo_ValidaOpcion()	23
4.4.3.21 vista_ErrorAperturaArchivo()	24
4.4.3.22 vista_ErrorEntrada()	24
4.4.3.23 vista_Menu()	24
4.4.3.24 vista_MenuDespliegue()	25
4.4.3.25 vista_MenuEpocas()	25
4.4.3.26 vista_MenuPeliculas()	25
4.4.3.27 vista_MenuRecomendacion()	25
4.4.3.28 vista_RegresaMenu()	25
<b>Índice alfabético</b>	<b>27</b>

# Capítulo 1

## Índice de clases

### 1.1. Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

<a href="#">matriz_struct</a>	.....	5
<a href="#">num_lineas</a>	.....	6



## Capítulo 2

# Indice de archivos

### 2.1. Lista de archivos

Lista de todos los archivos con descripciones breves:

<a href="#">DataShell.h</a>	7
<a href="#">Error.h</a>	12
<a href="#">Optimizacion.h</a>	13
<a href="#">Peliculas.h</a>	15





## Capítulo 3

# Documentación de las clases

### 3.1. Referencia de la Estructura matriz\_struct

```
#include <Películas.h>
```

#### Atributos públicos

- float \* [Datos](#) [BUFSIZ]
- size\_t [columnas](#)
- size\_t [filas](#)

#### 3.1.1. Documentación de los datos miembro

##### 3.1.1.1. columnas

```
size_t matriz_struct::columnas
```

##### 3.1.1.2. Datos

```
float* matriz_struct::Datos[BUFSIZ]
```

##### 3.1.1.3. filas

```
size_t matriz_struct::filas
```

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [Películas.h](#)

## 3.2. Referencia de la Estructura num\_lineas

```
#include <Peliculas.h>
```

### Atributos públicos

- `size_t` [columnas](#)
- `size_t` [filas](#)

### 3.2.1. Documentación de los datos miembro

#### 3.2.1.1. columnas

```
size_t num_lineas::columnas
```

#### 3.2.1.2. filas

```
size_t num_lineas::filas
```

La documentación para esta estructura fue generada a partir del siguiente fichero:

- [Películas.h](#)

## Capítulo 4

# Documentación de archivos

### 4.1. Referencia del Archivo DataShell.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Peliculas.h"
```

#### defines

- #define [EXTERN](#)

#### Funciones

- [EXTERN](#) void [vista\\_MenuDS](#) (void)  
*La funcion vista\_Menu pedira al usuario el nombre del archivo que desea ingresar al sistema para ser tokenizado y formateado.*
- [EXTERN](#) void [controlador\\_ProcesoDS](#) (char \*archivo, [MATRIZ](#) \*Matriz)  
*La funcion controlador\_Proceso controlará las llamadas a las funciones y las variables.*
- [EXTERN](#) FILE \* [modelo\\_Abre\\_Archivo](#) (char \*nombre\_archivo)  
*La funcion modelo\_Abre\_Archivo abrirá el archivo determinado con 'nombre\_archivo' en modo de lectura.*
- [EXTERN](#) int [modelo\\_Valida\\_Nombre](#) (char \*nombre\_archivo)  
*La funcion modelo\_Valida\_Nombre validará si el nombre de archivo ingresado por el usuario es correcto.*
- [EXTERN](#) void [modelo\\_Correccion\\_Nombre](#) (char \*nombre\_archivo)  
*La funcion modelo\_Correccion\_Nombre cambiará el leído por fgets por un \0.*
- [EXTERN](#) float \* [modelo\\_Tokenizer](#) (char \*linea, float \*buffer)  
*La funcion modelo\_Tokenizer separara la cadena en cada coma.*
- [EXTERN](#) size\_t [modelo\\_Columnas](#) (char \*linea)  
*La funcion modelo\_Columnas nos permite calcular la cantidad de columnas presentes en el archivo leído.*
- [EXTERN](#) void [vista\\_Error\\_Menos1](#) (void)  
*La funcion vista\_Error\_Menos1 mostrará un mensaje de error indicando que debería hacer para corregirlo.*
- [EXTERN](#) void [vista\\_Error\\_Apertura\\_Archivo](#) (void)  
*La funcion vista\_Error\_Apertura\_Archivo mostrará un mensaje de error indicando que debería hacer para corregirlo.*
- [EXTERN](#) char \* [controlador\\_linea](#) (char fila[])  
*La funcion controlador\_linea quita las comas de la linea y los sustituye por espacios.*

### 4.1.1. Documentación de los 'defines'

#### 4.1.1.1. EXTERN

```
#define EXTERN
```

[DataShell.h](#)

Creado por Rodrigo Garcia Diaz y Jesus Enrique Domínguez el 27 de octubre del 2020. System headers required by the following declarations (the implementation will import its specific dependencies): Application specific headers required by the following declarations (the implementation will import its specific dependencies):

### 4.1.2. Documentación de las funciones

#### 4.1.2.1. controlador\_linea()

```
EXTERN char* controlador_linea (
    char fila[] )
```

La funcion controlador\_linea quita las comas de la linea y los sustituye por espacios.

##### Parámetros

<i>fila</i>	(char *): cadena que se sustituirá las comas.
-------------	---

##### Devuelve

apuntador a al arreglo que se edito.

#### 4.1.2.2. controlador\_ProcesoDS()

```
EXTERN void controlador_ProcesoDS (
    char * archivo,
    MATRIZ * Matriz )
```

La funcion controlador\_Proceso controlará las llamadas a las funciones y las variables.

##### Parámetros

<i>archivo</i>	(char *): nombre del archivo que se abrirá para el correspondiente proceso
----------------	--

**Devuelve**

void

**4.1.2.3. modelo\_Abre\_Archivo()**

```
EXTERN FILE* modelo_Abre_Archivo (  
    char * nombre_archivo )
```

La funcion modelo\_Abre\_Archivo abrirà el archivo determinado con 'nombre\_archivo' en modo de lectura.

**Parámetros**

<i>nombre_archivo</i>	(char *): nombre del archivo que se abrirà para el correspondiente proceso
-----------------------	--

**Devuelve**

Archivo abierto en el modo de lectura

**4.1.2.4. modelo\_Columnas()**

```
EXTERN size_t modelo_Columnas (  
    char * linea )
```

La funcion modelo\_Columnas nos permite calcular la cantidad de columnas presentes en el archivo leído.

**Parámetros**

<i>linea</i>	(char *): cadena con la que se calcularàn las columnas presentes
--------------	--

**Devuelve**

cantidad de columnas en el archivo

**4.1.2.5. modelo\_Correccion\_Nombre()**

```
EXTERN void modelo_Correccion_Nombre (  
    char * nombre_archivo )
```

La funcion modelo\_Correccion\_Nombre cambiarà el leído por fgets por un \0.

**Parámetros**

<i>nombre_archivo</i>	(char *): nombre del archivo a corregir
-----------------------	---

**Devuelve**

none

**4.1.2.6. modelo\_Tokenizer()**

```
EXTERN float* modelo_Tokenizer (  
    char * linea,  
    float * buffer )
```

La funcion modelo\_Tokenizer separara la cadena en cada coma.

**Parámetros**

<i>linea</i>	(char *): cadena que va a separar
--------------	-----------------------------------

**Devuelve**

puntero con al arreglo de datos separados para ser metidos a la matriz de datos.

**4.1.2.7. modelo\_Valida\_Nombre()**

```
EXTERN int modelo_Valida_Nombre (  
    char * nombre_archivo )
```

La funcion modelo\_Valida\_Nombre validará si el nombre de archivo ingresado por el usuario es correcto.

**Parámetros**

<i>nombre_archivo</i>	(char *): nombre del archivo que se validará
-----------------------	--

**Devuelve**

Entero de confirmación de si es correcto o no

#### 4.1.2.8. vista\_Error\_Apertura\_Archivo()

```
EXTERN void vista_Error_Apertura_Archivo (  
    void )
```

La funcion vista\_Error\_Apertura\_Archivo mostrarà un mensaje de error indicando que deberia hacer para corregirlo.

##### Parámetros

<i>none</i>	
-------------	--

##### Devuelve

*none*

#### 4.1.2.9. vista\_Error\_Menos1()

```
EXTERN void vista_Error_Menos1 (  
    void )
```

La funcion vista\_Error\_Menos1 mostrarà un mensaje de error indicando que deberia hacer para corregirlo.

##### Parámetros

<i>none</i>	
-------------	--

##### Devuelve

*none*

#### 4.1.2.10. vista\_MenuDS()

```
EXTERN void vista_MenuDS (  
    void )
```

La funcion vista\_Menu pedira al usuario el nombre del archivo que desea ingresar al sistema para ser tokenizado y formateado.

##### Parámetros

<i>void</i>	
-------------	--

Devuelve

none

## 4.2. Referencia del Archivo Error.h

```
#include <stdio.h>
#include <stdlib.h>
```

### defines

- #define [EXTERN](#)

### Funciones

- [EXTERN](#) void [controlador\\_errores](#) (int main, float error, char \*modo)  
*La funcion controlador\_errores inserta el nuevo error a un archivo que contiene los errores anteriores.*
- [EXTERN](#) float [controlador\\_errorpromedio](#) ()  
*La funcion controlador\_errorpromedio calcula el error promedio del archivo error.r.*

### 4.2.1. Documentación de los 'defines'

#### 4.2.1.1. EXTERN

```
#define EXTERN
```

[Error.h](#)

Creado por Rodrigo Garcia Diaz y Jesus Enrique Domínguez el 21 de Noviembre del 2020.

### 4.2.2. Documentación de las funciones

#### 4.2.2.1. controlador\_errores()

```
EXTERN void controlador_errores (
    int main,
    float error,
    char * modo )
```

La funcion controlador\_errores inserta el nuevo error a un archivo que contiene los errores anteriores.



## Parámetros

<code>error</code>	(float): Es el valor del error actual para insertar en el archivo
--------------------	---

## Devuelve

none

4.2.2.2. `controlador_errorpromedio()`

```
EXTERN float controlador_errorpromedio ( )
```

La funcion `controlador_errorpromedio` calcula el error promedio del archivo `error.r`.

## Devuelve

float

## 4.3. Referencia del Archivo Optimizacion.h

```
#include <stdio.h>
#include <stdlib.h>
#include "Peliculas.h"
```

## defines

- `#define EXTERN`

## Funciones

- `EXTERN float * modelo_optimizacion_content` (float \*user, float \*content, size\_t size, float n, float riu)  
*La funcion `modelo_optimizacion_content` obtiene los nuevos vectores optimizados para los items, regresa el vector correspondiente.*
- `EXTERN void modelo_optimizacion_user` (MATRIZ \*User, MATRIZ \*Content, size\_t user\_index, size\_t movie, float error, long double n)  
*La funcion `modelo_optimizacion_content` obtiene los nuevos vectores optimizados para el user, regresa el vector correspondiente.*
- `EXTERN float * controlador_transpuesta` (MATRIZ \*matriz, int id, float \*vector)  
*La funcion `controlador_transpuesta` obtiene los nuevos vectores que convierten una columna a su transpuesta. Regresa el vector de tipo float.*

## 4.3.1. Documentación de los 'defines'

#### 4.3.1.1. EXTERN

```
#define EXTERN
```

### 4.3.2. Documentación de las funciones

#### 4.3.2.1. controlador\_transpuesta()

```
EXTERN float* controlador_transpuesta (
    MATRIZ * matriz,
    int id,
    float * vector )
```

La funcion controlador\_transpuesta obtiene los nuevos vectores que convierten una columna a su transpuesta. Regresa el vector de tipo float.

##### Parámetros

<i>matriz</i>	(*MATRIZ): estructura con matriz y dimensiones
---------------	--

id (int): indice del item o user que sera una constante;

##### Devuelve

float

#### 4.3.2.2. modelo\_optimizacion\_content()

```
EXTERN float* modelo_optimizacion_content (
    float * user,
    float * content,
    size_t size,
    float n,
    float riu )
```

La funcion modelo\_optimizacion\_content obtiene los nuevos vectores optimizados para los items, regresa el vector correspondiente.

##### Parámetros

<i>user</i>	(*float): Vector particular de valores desde la matriz de userdb
-------------	--

content (\*float): Vector particular de valores desde la matriz de contentdb

size (size\_t): Valor que indica el tamaño de los vectores

n (float): Valor de la n

riu (float): Valor del rating

**Devuelve**

float

#### 4.3.2.3. modelo\_optimizacion\_user()

```
EXTERN void modelo_optimizacion_user (
    MATRIZ * User,
    MATRIZ * Content,
    size_t user_index,
    size_t movie,
    float error,
    long double n )
```

La funcion modelo\_optimizacion\_content obtiene los nuevos vectores optimizados para el user, regresa el vector correspondiente.

**Parámetros**

<i>user</i>	(*float): Vector particular de valores desde la matriz de userdb
-------------	--

content (\*float): Vector particular de valores desde la matriz de contentdb

size (size\_t): Valor que indica el tamaño de los vectores

n (float): Valor de la n

riu (float): Valor del rating

**Devuelve**

float

## 4.4. Referencia del Archivo Peliculas.h

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
```

## Clases

- struct `num_lineas`
- struct `matriz_struct`

## defines

- `#define` `EXTERN`

## typedefs

- typedef struct `num_lineas` `TAMANO`
- typedef struct `matriz_struct` `MATRIZ`

## Funciones

- `EXTERN` void `controlador_Proceso` (int new\_file, char \*opcion)  
*La funcion controlador\_Proceso mantiene las funcionalidades, settea las variables y llama a las funciones determinadas en momentos adecuadas para la continuidad del programa.*
- `EXTERN` void `vista_Menu` (int new\_file)  
*La funcion vista\_Menu es la función inicial que dará la bienvenida y comenzara el programa.*
- `EXTERN` float \* `modelo_IniciaUserDB` (size\_t columnas, float \*buffer)  
*La funcion modelo\_IniciaUserDB inicializará de forma aleatoria la matriz de afinidades de los usuarios con las categorías presentes en Categorías.txt. Este proceso solo se hará al iniciar por primera vez el algoritmo.*
- `EXTERN` float `modelo_Prediccion` (float \*P\_user, `MATRIZ` \*Q\_item, int index)  
*La funcion modelo\_Prediccion calcula el intento inicial entre P[user] y Q[item].*
- `EXTERN` float `modelo_Error` (float \*prediccion, `MATRIZ` \*Rating, int user, int item)  
*La funcion modelo\_Error calcula el error generado por la predicción del sistema.*
- `EXTERN` void `modelo_ImprimeUserDB` (float \*Afinidades[], size\_t filas, size\_t columnas)  
*La funcion modelo\_ImprimeUserDB creará un archivo CSV con los datos en la matriz de Afinidades.*
- `EXTERN` void `modelo_DeterminaTamano` (`TAMANO` \*dimensiones)  
*La funcion modelo\_DeterminaTamano devolverá la cantidad de líneas de un archivo dado. Con esto, se podrá determinar cuantos números aleatorios se eben ingresar.*
- `EXTERN` void `modelo_ImprimeRanking` (`MATRIZ` \*RankingDB)  
*La funcion modelo\_ImprimeRanking imprimirá una representación de la tabla RankingDB de los usuarios.*
- `EXTERN` int `modelo_ValidaOpcion` (char \*opcion)  
*La funcion modelo\_ValidaOpcion valida si un string solo contiene valores numéricos.*
- `EXTERN` FILE \* `modelo_AbreArchivo` (char \*nombre\_archivo, char \*modo)  
*La funcion modelo\_AbreArchivo, abrirá y regresará el pointer a un file, así mismo, validará si la apertura fue correcta o no, indicando si exitio algún problema.*
- `EXTERN` size\_t `modelo_RecorreTXT` (FILE \*archivo)  
*La funcion modelo\_RecorreTXT lee las líneas de los archivos UserNames.txt y MovieNames.txt para determinar el tamaño de la matriz a crear.*
- `EXTERN` void `modelo_CreaUserDB` (`TAMANO` \*dimensiones)  
*La funcion modelo\_UserDB crea un archivo con la nueva matriz de afinidades de los usuarios.*
- `EXTERN` char \* `modelo_ObtienLinea` (char \*linea\_leida, int index, char \*nombre\_archivo)
- `EXTERN` void `modelo_DespliegaUsuarios` (`MATRIZ` \*Ranking, char \*opcion)
- `EXTERN` void `modelo_DespliegaPeliculas` (`MATRIZ` \*Ranking, char \*opcion)
- `EXTERN` float `modelo_CalculaPromedio` (`MATRIZ` \*Ranking, int movie)
- `EXTERN` void `modelo_DotRecomendacion` (`MATRIZ` \*Ranking, float \*usuario, `MATRIZ` \*Contenido, int user, char \*opcion)

- **EXTERN** void `modelo_DotSugerencia` (**MATRIZ** \*User, float \*usuario, int user, char \*opcion)
- **EXTERN** void `modelo_DotSimilares` (**MATRIZ** \*Content, int movie, char \*opcion)
- **EXTERN** void `modelo_GraficaError` (char \*archivo)
- **EXTERN** void `DataShell` (char \*archivo, **MATRIZ** \*Matriz)  
*La función `DataShell()` realiza las funciones previamente realizadas en la practica del `DataShell`.*
- **EXTERN** void `vista_ErrorAperturaArchivo` (void)  
*La función `vista_Error_Apertura_Archivo` mostrará un mensaje de error indicando que debería hacer para corregirlo.*
- **EXTERN** void `vista_ErrorEntrada` (int new\_file, int retorno, char \*despliegue)  
*La función `vista_ErrorEntrada` mostrará un mensaje para cuando la entrada en el menú es incorrecta.*
- **EXTERN** void `vista_RegresaMenu` (int new\_file, int retorno, char \*opcion)
- **EXTERN** char \* `vista_MenuDespliegue` (void)
- **EXTERN** char \* `vista_MenuEpocas` (void)
- **EXTERN** char \* `vista_MenuRecomendacion` (**MATRIZ** \*Usuario, int modo)
- **EXTERN** char \* `vista_MenuPeliculas` (**MATRIZ** \*CContenido)

#### 4.4.1. Documentación de los 'defines'

##### 4.4.1.1. EXTERN

```
#define EXTERN
```

`Peliculas.h`

Creado por Rodrigo Garcia Diaz y Jesus Enrique Domínguez el 21 de Noviembre del 2020.

#### 4.4.2. Documentación de los 'typedefs'

##### 4.4.2.1. MATRIZ

```
typedef struct matriz_struct MATRIZ
```

##### 4.4.2.2. TAMANO

```
typedef struct num_lineas TAMANO
```

#### 4.4.3. Documentación de las funciones

##### 4.4.3.1. controlador\_Proceso()

```
EXTERN void controlador_Proceso (  
    int new_file,  
    char * opcion )
```

La función `controlador_Proceso` mantiene las funcionalidades, settea las variables y llama a las funciones determinadas en momentos adecuadas para la continuidad del programa.

**Parámetros**

<i>new_file</i>	(int): Este valor determina si se usará una UserDB existente o si creará una nueva mediante el comando '-new'
-----------------	---

**Devuelve**

none

**4.4.3.2. DataShell()**

```
EXTERN void DataShell (
    char * archivo,
    MATRIZ * Matriz )
```

La función [DataShell\(\)](#) realiza las funciones previamente realizadas en la practica del DataShell.

**Parámetros**

<i>archivo</i>	(char *): archivo de donde obtener los datos para llenar las matrices.
----------------	--

**Devuelve**

numero de columnas/filas.

**4.4.3.3. modelo\_AbreArchivo()**

```
EXTERN FILE* modelo_AbreArchivo (
    char * nombre_archivo,
    char * modo )
```

La función `modelo_AbreArchivo`, abrirá y regresará el pointer a un file, así mismo, validará si la apertura fue correcta o no, indicando si exitio algún problema.

**Parámetros**

<i>nombre_archivo</i>	(char *)
-----------------------	----------

**Devuelve**

puntero FILE con el archivo abierto

**4.4.3.4. modelo\_CalculaPromedio()**

```
EXTERN float modelo_CalculaPromedio (
    MATRIZ * Ranking,
    int movie )
```

**4.4.3.5. modelo\_CreaUserDB()**

```
EXTERN void modelo_CreaUserDB (
    TAMANO * dimensiones )
```

La funcion modelo\_UserDB crea un archivo con la nueva matriz de afinidades de los usuarios.

**Parámetros**

<i>dimensiones</i>	(TAMANO *) variable para determinar los tamaños de la matriz nueva
--------------------	--

**Devuelve**

none

**4.4.3.6. modelo\_DespliegaPeliculas()**

```
EXTERN void modelo_DespliegaPeliculas (
    MATRIZ * Ranking,
    char * opcion )
```

**4.4.3.7. modelo\_DespliegaUsuarios()**

```
EXTERN void modelo_DespliegaUsuarios (
    MATRIZ * Ranking,
    char * opcion )
```

**4.4.3.8. modelo\_DeterminaTamano()**

```
EXTERN void modelo_DeterminaTamano (
    TAMANO * dimensiones )
```

La funcion modelo\_DeterminaTamano devolverà la cantidad de lineas de un archivo dado. Con esto, se podrá determinar cuantos números aleatorios se eben ingresar.

**Parámetros**

<i>dimensiones</i>	(TAMANO *)
--------------------	------------

**Devuelve**

none

**4.4.3.9. modelo\_DotRecomendacion()**

```
EXTERN void modelo_DotRecomendacion (
    MATRIZ * Ranking,
    float * usuario,
    MATRIZ * Contenido,
    int user,
    char * opcion )
```

**4.4.3.10. modelo\_DotSimilares()**

```
EXTERN void modelo_DotSimilares (
    MATRIZ * Content,
    int movie,
    char * opcion )
```

**4.4.3.11. modelo\_DotSugerencia()**

```
EXTERN void modelo_DotSugerencia (
    MATRIZ * User,
    float * usuario,
    int user,
    char * opcion )
```

**4.4.3.12. modelo\_Error()**

```
EXTERN float modelo_Error (
    float * prediccion,
    MATRIZ * Rating,
    int user,
    int item )
```

La funcion `modelo_Error` calcula el error generado por la prediccion del sistema.



## Parámetros

<i>Rating</i>	(MATRIZ *): RankingDB P_user (float *): Pointer que apunta al valor generado por el modulo de la prediccion. index (int): índice de la película que se está analizando; user (int): índice del usuario que se está analizando
---------------	---

## Devuelve

valor del error

## 4.4.3.13. modelo\_GraficaError()

```
EXTERN void modelo_GraficaError (
    char * archivo )
```

## 4.4.3.14. modelo\_ImprimeRanking()

```
EXTERN void modelo_ImprimeRanking (
    MATRIZ * RankingDB )
```

La funcion modelo\_ImprimeRanking imprimirà una representaciòn de la tabla RankingDB de los usuarios.

## Parámetros

<i>RankingDB</i>	(MATRIZ *)
------------------	------------

## Devuelve

none

## 4.4.3.15. modelo\_ImprimeUserDB()

```
EXTERN void modelo_ImprimeUserDB (
    float * Afinidades[],
    size_t filas,
    size_t columnas )
```

La funcion modelo\_ImprimeUserDB crearà un archivo CSV con los datos en la matriz de Afinidades.

**Parámetros**

<i>columnas</i>	(size_t): tamaño de columnas para imprimir la matriz en el archivo Afinidades[] (float *) Matriz que se imprimirá en el archivo .csv
-----------------	--

**Devuelve**

array con los valores aleatorios

**4.4.3.16. modelo\_IniciaUserDB()**

```
EXTERN float* modelo_IniciaUserDB (
    size_t columnas,
    float * buffer )
```

La funcion modelo\_IniciaUserDB inicializarà de forma aleatoria la matriz de afinidades de los usuarios con las categorias presentes en Categorias.txt. Este proceso solo se harà al iniciar por primera vez el algoritmo.

**Parámetros**

<i>columnas</i>	(size_t): tamaño de columnas para crear el array buffer (float *) memoria en donde se guardará el arreglo de datos.
-----------------	---

**Devuelve**

array con los valores aleatorios

**4.4.3.17. modelo\_ObtenerLinea()**

```
EXTERN char* modelo_ObtenerLinea (
    char * linea_leida,
    int index,
    char * nombre_archivo )
```

**4.4.3.18. modelo\_Prediccion()**

```
EXTERN float modelo_Prediccion (
    float * P_user,
    MATRIZ * Q_item,
    int index )
```

La funcion modelo\_Prediccion calcula el intento inicial entre P[user] y Q[item].

## Parámetros

<i>Q_Item</i>	(MATRIZ *): ContentDB P_user (float *): Arreglo con las afinidades del usuario dado por un contador. index (int): índice de la película que se está analizando.
---------------	---

## Devuelve

valor del intento inicial

**4.4.3.19. modelo\_RecorreTXT()**

```
EXTERN size_t modelo_RecorreTXT (
    FILE * archivo )
```

La función `modelo_RecorreTXT` lee las líneas de los archivos `UserNames.txt` y `MovieNames.txt` para determinar el tamaño de la matriz a crear.

## Parámetros

<i>archivo</i>	(FILE *): archivo a recorrer. dimensiones (TAMANO *) variable donde guardar los tamaños
----------------	---

## Devuelve

número de columnas/filas.

**4.4.3.20. modelo\_ValidaOpcion()**

```
EXTERN int modelo_ValidaOpcion (
    char * opcion )
```

La función `modelo_ValidaOpcion` valida si un string solo contiene valores numéricos.

## Parámetros

<i>opcion</i>	(char *)
---------------	----------

## Devuelve

none

#### 4.4.3.21. vista\_ErrorAperturaArchivo()

```
EXTERN void vista_ErrorAperturaArchivo (
    void )
```

La funcion vista\_Error\_Apertura\_Archivo mostrarà un mensaje de error indicando que deberìa hacer para corregirlo.

##### Parámetros

<i>none</i>	
-------------	--

##### Devuelve

*none*

#### 4.4.3.22. vista\_ErrorEntrada()

```
EXTERN void vista_ErrorEntrada (
    int new_file,
    int retorno,
    char * despliegue )
```

La funcion vista\_ErrorEntrada mostrarà un mensaje para cuando la entrada en el menù es incorrecta.

##### Parámetros

<i>none</i>	
-------------	--

##### Devuelve

*none*

#### 4.4.3.23. vista\_Menu()

```
EXTERN void vista_Menu (
    int new_file )
```

La funcion vista\_Menu es la funciòn inicial que darà la bienvenida y comenzara el programa.

##### Parámetros

<i>new_file</i>	(int): Este valor determina si se usará una UserDB existente o si creará una nueva mediante el comando '-new'
-----------------	---

Devuelve

none

#### 4.4.3.24. vista\_MenuDespliegue()

```
EXTERN char* vista_MenuDespliegue (
    void )
```

#### 4.4.3.25. vista\_MenuEpocas()

```
EXTERN char* vista_MenuEpocas (
    void )
```

#### 4.4.3.26. vista\_MenuPeliculas()

```
EXTERN char* vista_MenuPeliculas (
    MATRIZ * CContenido )
```

#### 4.4.3.27. vista\_MenuRecomendacion()

```
EXTERN char* vista_MenuRecomendacion (
    MATRIZ * Usuario,
    int modo )
```

#### 4.4.3.28. vista\_RegresaMenu()

```
EXTERN void vista_RegresaMenu (
    int new_file,
    int retorno,
    char * opcion )
```



# Índice alfabético

columnas  
    matriz\_struct, 5  
    num\_lineas, 6  
controlador\_errores  
    Error.h, 12  
controlador\_errorpromedio  
    Error.h, 13  
controlador\_linea  
    DataShell.h, 8  
controlador\_Proceso  
    Películas.h, 17  
controlador\_ProcesoDS  
    DataShell.h, 8  
controlador\_transpuesta  
    Optimizacion.h, 14  
  
DataShell  
    Películas.h, 18  
DataShell.h, 7  
    controlador\_linea, 8  
    controlador\_ProcesoDS, 8  
    EXTERN, 8  
    modelo\_Abre\_Archivo, 9  
    modelo\_Columnas, 9  
    modelo\_Correccion\_Nombre, 9  
    modelo\_Tokenizer, 10  
    modelo\_Valida\_Nombre, 10  
    vista\_Error\_Apertura\_Archivo, 10  
    vista\_Error\_Menos1, 11  
    vista\_MenuDS, 11  
Datos  
    matriz\_struct, 5  
  
Error.h, 12  
    controlador\_errores, 12  
    controlador\_errorpromedio, 13  
    EXTERN, 12  
EXTERN  
    DataShell.h, 8  
    Error.h, 12  
    Optimizacion.h, 13  
    Películas.h, 17  
  
filas  
    matriz\_struct, 5  
    num\_lineas, 6  
  
MATRIZ  
    Películas.h, 17  
matriz\_struct, 5  
  
columnas, 5  
Datos, 5  
filas, 5  
modelo\_Abre\_Archivo  
    DataShell.h, 9  
modelo\_AbreArchivo  
    Películas.h, 18  
modelo\_CalculaPromedio  
    Películas.h, 18  
modelo\_Columnas  
    DataShell.h, 9  
modelo\_Correccion\_Nombre  
    DataShell.h, 9  
modelo\_CreaUserDB  
    Películas.h, 19  
modelo\_DespliegaPelículas  
    Películas.h, 19  
modelo\_DespliegaUsuarios  
    Películas.h, 19  
modelo\_DeterminaTamano  
    Películas.h, 19  
modelo\_DotRecomendacion  
    Películas.h, 20  
modelo\_DotSimilares  
    Películas.h, 20  
modelo\_DotSugerencia  
    Películas.h, 20  
modelo\_Error  
    Películas.h, 20  
modelo\_GraficaError  
    Películas.h, 21  
modelo\_ImprimeRanking  
    Películas.h, 21  
modelo\_ImprimeUserDB  
    Películas.h, 21  
modelo\_IniciaUserDB  
    Películas.h, 22  
modelo\_ObteneLinea  
    Películas.h, 22  
modelo\_optimizacion\_content  
    Optimizacion.h, 14  
modelo\_optimizacion\_user  
    Optimizacion.h, 15  
modelo\_Prediccion  
    Películas.h, 22  
modelo\_RecorreTXT  
    Películas.h, 23  
modelo\_Tokenizer  
    DataShell.h, 10

modelo\_Valida\_Nombre  
    DataShell.h, 10  
modelo\_ValidaOpcion  
    Peliculas.h, 23  
  
num\_lineas, 6  
    columnas, 6  
    filas, 6  
  
Optimizacion.h, 13  
    controlador\_transpuesta, 14  
    EXTERN, 13  
    modelo\_optimizacion\_content, 14  
    modelo\_optimizacion\_user, 15  
  
Peliculas.h, 15  
    controlador\_Proceso, 17  
    DataShell, 18  
    EXTERN, 17  
    MATRIZ, 17  
    modelo\_AbreArchivo, 18  
    modelo\_CalculaPromedio, 18  
    modelo\_CreaUserDB, 19  
    modelo\_DespliegaPeliculas, 19  
    modelo\_DespliegaUsuarios, 19  
    modelo\_DeterminaTamano, 19  
    modelo\_DotRecomendacion, 20  
    modelo\_DotSimilares, 20  
    modelo\_DotSugerencia, 20  
    modelo\_Error, 20  
    modelo\_GraficaError, 21  
    modelo\_ImprimeRanking, 21  
    modelo\_ImprimeUserDB, 21  
    modelo\_IniciaUserDB, 22  
    modelo\_ObtienLinea, 22  
    modelo\_Prediccion, 22  
    modelo\_RecorreTXT, 23  
    modelo\_ValidaOpcion, 23  
    TAMANO, 17  
    vista\_ErrorAperturaArchivo, 23  
    vista\_ErrorEntrada, 24  
    vista\_Menu, 24  
    vista\_MenuDespliegue, 25  
    vista\_MenuEpocas, 25  
    vista\_MenuPeliculas, 25  
    vista\_MenuRecomendacion, 25  
    vista\_RegresaMenu, 25  
  
TAMANO  
    Peliculas.h, 17  
  
vista\_Error\_Apertura\_Archivo  
    DataShell.h, 10  
vista\_Error\_Menos1  
    DataShell.h, 11  
vista\_ErrorAperturaArchivo  
    Peliculas.h, 23  
vista\_ErrorEntrada  
    Peliculas.h, 24  
  
vista\_Menu  
    Peliculas.h, 24  
vista\_MenuDespliegue  
    Peliculas.h, 25  
vista\_MenuDS  
    DataShell.h, 11  
vista\_MenuEpocas  
    Peliculas.h, 25  
vista\_MenuPeliculas  
    Peliculas.h, 25  
vista\_MenuRecomendacion  
    Peliculas.h, 25  
vista\_RegresaMenu  
    Peliculas.h, 25