

Facultad de Ingenierías en Electricidad y Computación



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

2015 - 2016

|  |  |
| --- | --- |
| http://ocw.upm.es/lenguajes-y-sistemas-informaticos/estructuras-de-datos/contenidos/estructurasDeDatos.jpg  Estructura de datos | aLUMNos:  guillermo bernal  Carlos cedeño  bryan Ordóez  Paralelo: 2  Docente: Ing. Jorge Rodríguez Echeverría  I PARCIAL – II TÉRMINO |

# **Identifique los TDA’s presentes en el proyecto y explique porque los creó.**

Entre los TDA’s presentes en nuestro proyecto, hay que hacer una diferenciación entre los cuales fueron creados y cuales fueron implementados.

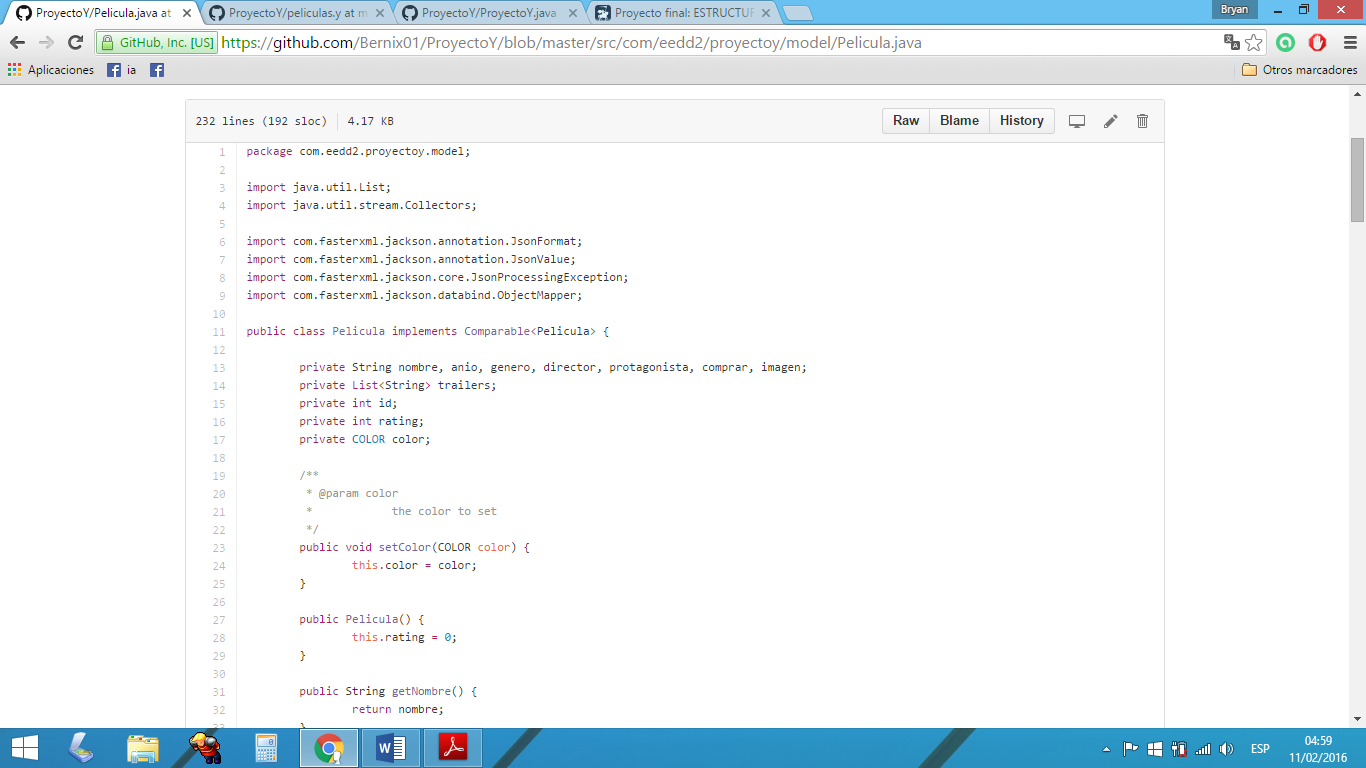
De los TDA’s que fueron creados hay solo 1:

* Película:

El TDA implementa Comparable por ID. Este Comparable nos ayudará mucho luego cuando comencemos a ordenar los datos para el debido funcionamiento de la aplicación del proyecto y para la eficiencia del mismo, por ahora solo señalaremos que está.

Las características o parámetros que forman parte de Pelicula son:

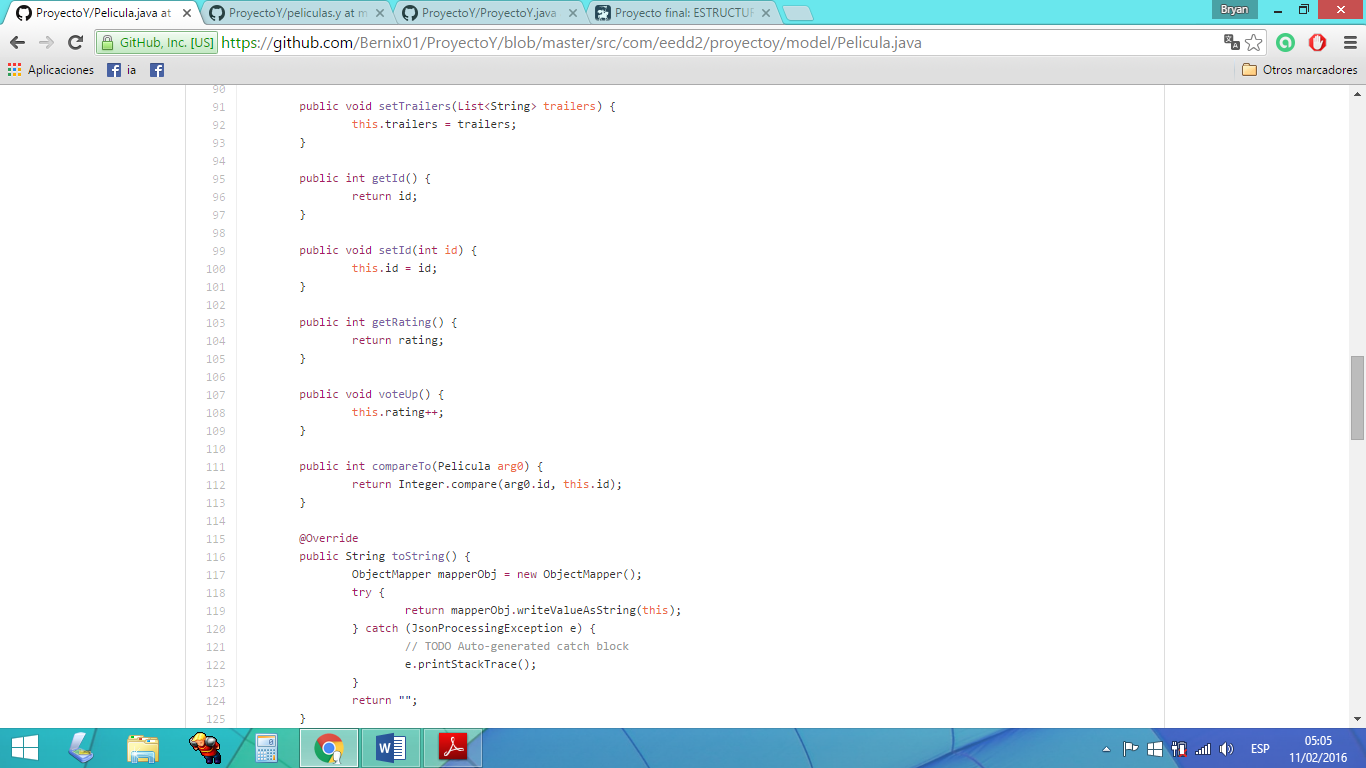
* **ID:** A cada película le corresponde un ID único, lo cual nos ayudará si existen películas con el mismo nombre.
* **Nombre:** Corresponde al nombre de la película, la cual nos permite ubicarla y buscarla entre las demás.
* **Año(Anio):** Contiene el año en que una película fue estrenada.
* **Género:** Contiene el género o tipo de película. Ej: Terror, Comedio, Romance, Sci-fi, etc.
* **Director:** Este campo corresponde al nombre de la persona que dirigió la película.
* **Protagonista:** Este campo corresponde al nombre del protagonista de la película. Los campos director y protagonista serán de importancia en el momento de implementar el proyecto.
* **Comprar:** Contiene el link correspondiente para poder comprar una película en amazon.
* **Tráiler:** Contiene la lista de tráileres disponibles de la película.
* **Imagen:** Es la imagen de portada de la película. Es lo primero que se verá de una película además de su nombre.
* **Rating**: Contiene el número de visitas que los usuarios han realizado a una película. Se ha inicializado el rating = 0, para que así, cada vez que una película es seleccionada, aumente en 1.



Ahora, los métodos presentes en el TDA película son:

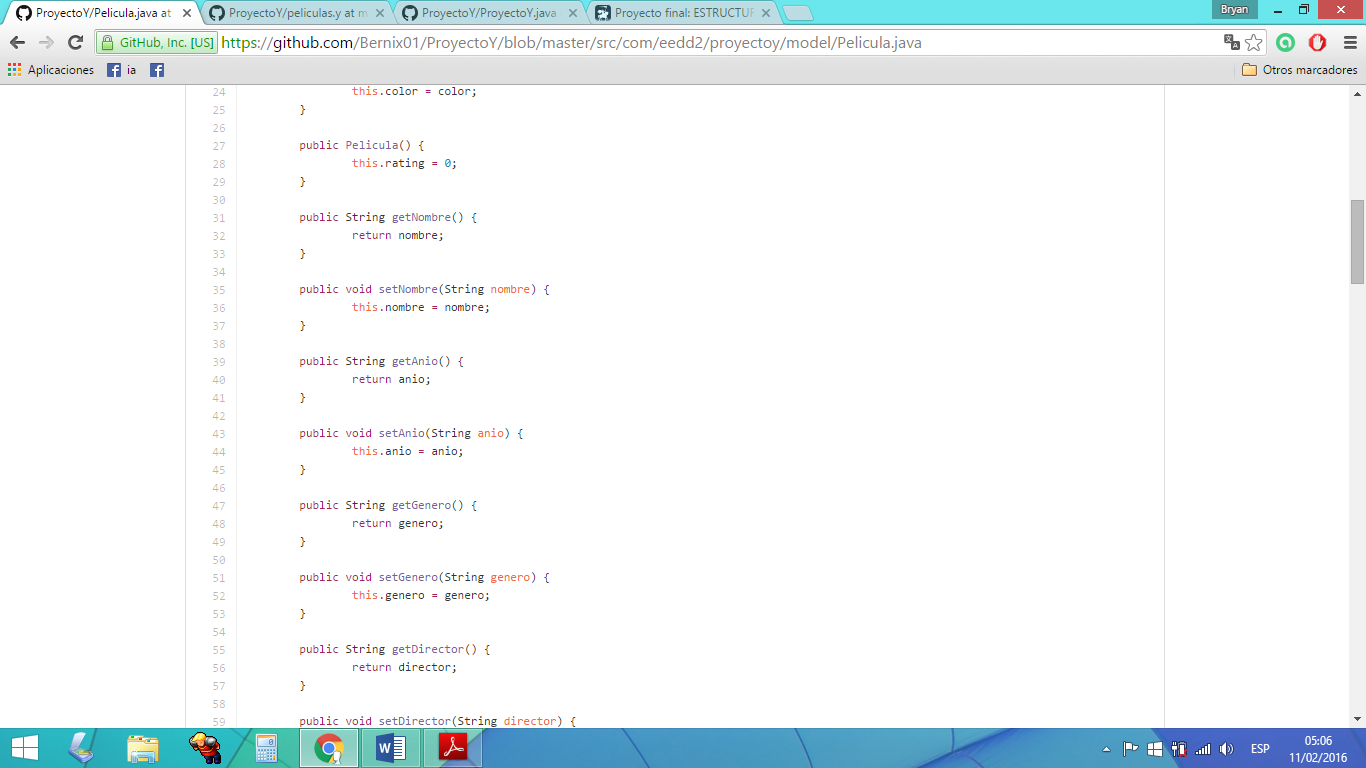
* Get/SetId()

El método Get permite obtener el dato privado Id de película. El método Set permite cambiar el valor de la variable Id



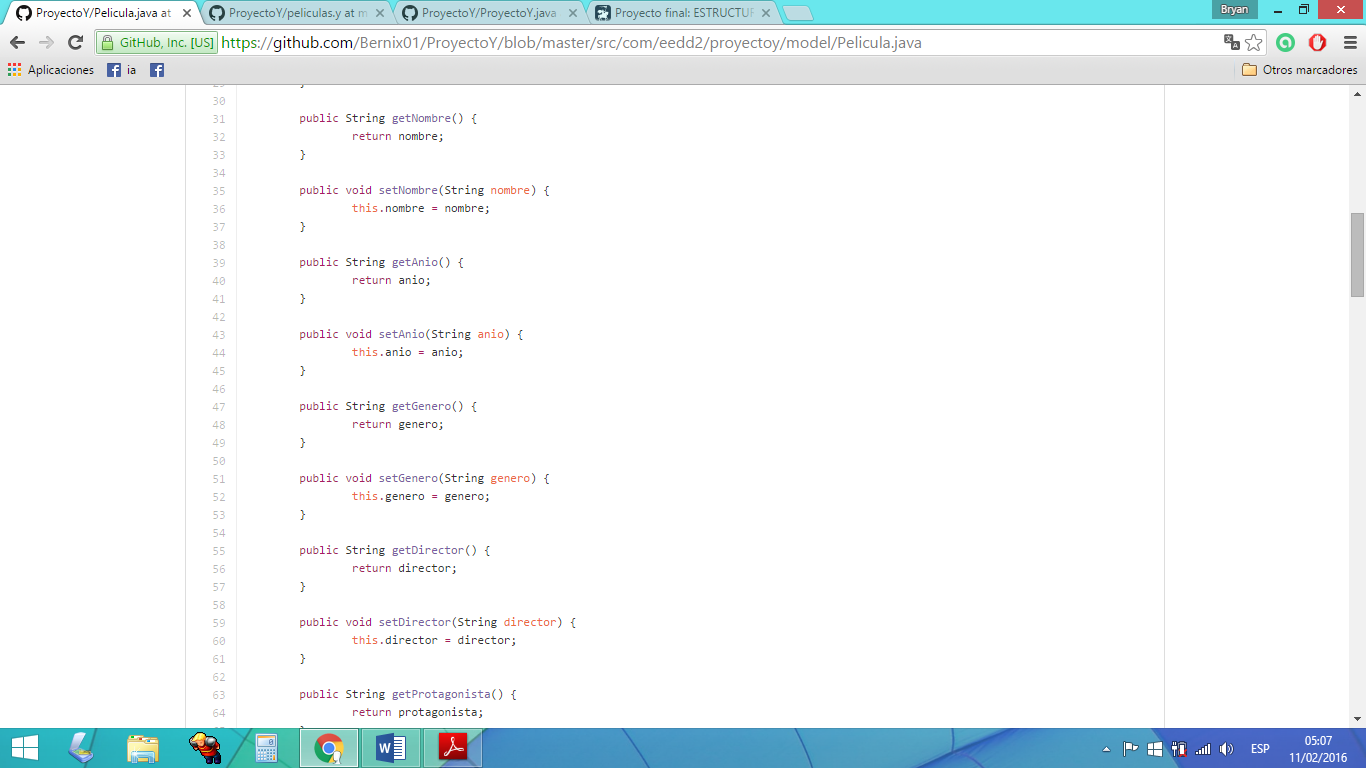
* Get/SetAnio()

El método Get permite obtener el dato privado Anio de película. El método Set permite cambiar el valor de la variable Anio.



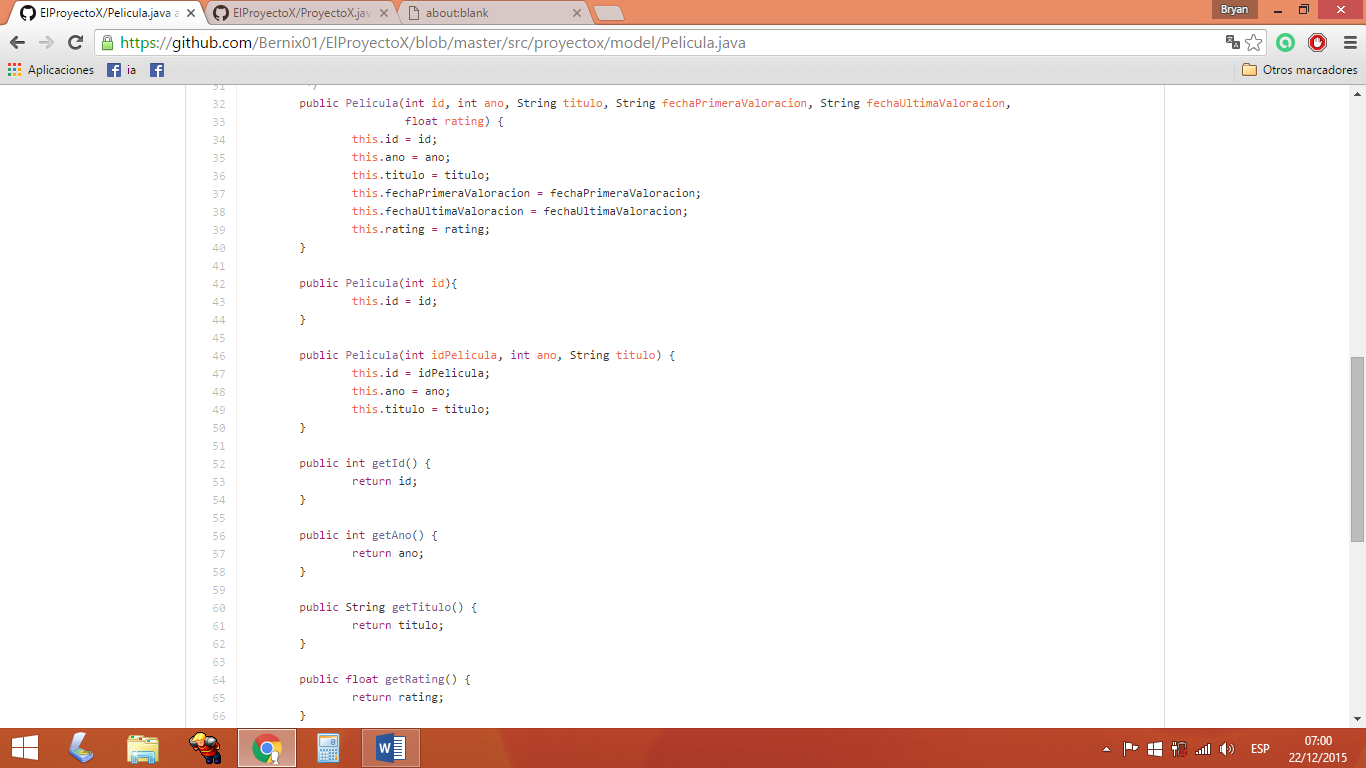
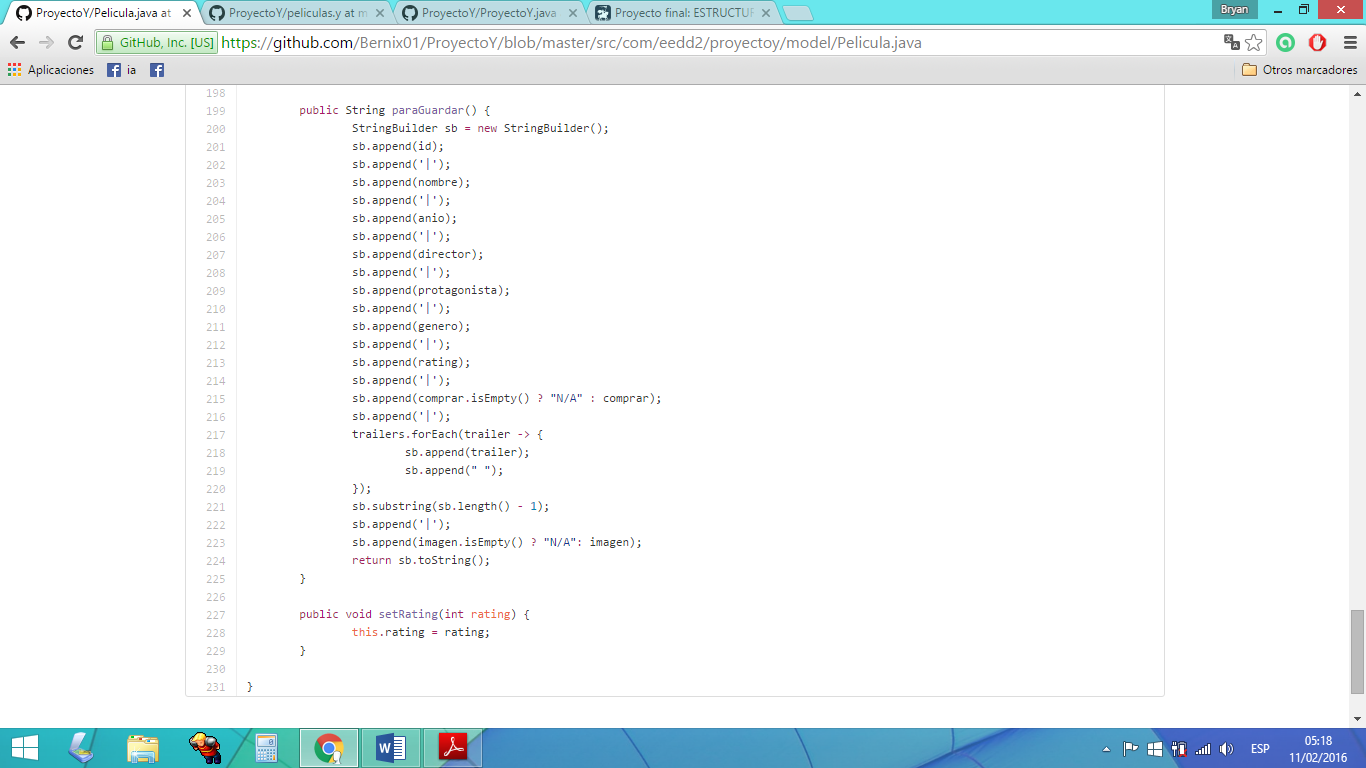
* Get/SetNombre()

El método Get permite obtener el dato privado título de película. El método Set permite cambiar el valor nombre, es decir cambiar de nombre a la película.



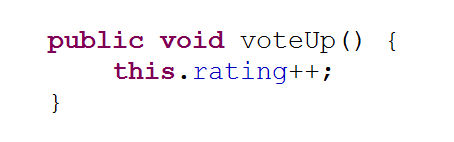
* GetSetRating()

El método Get permite obtener el dato privado Rating dentro de película. El método Set permite cambiar el valor del dato privado Rating dentro de película.

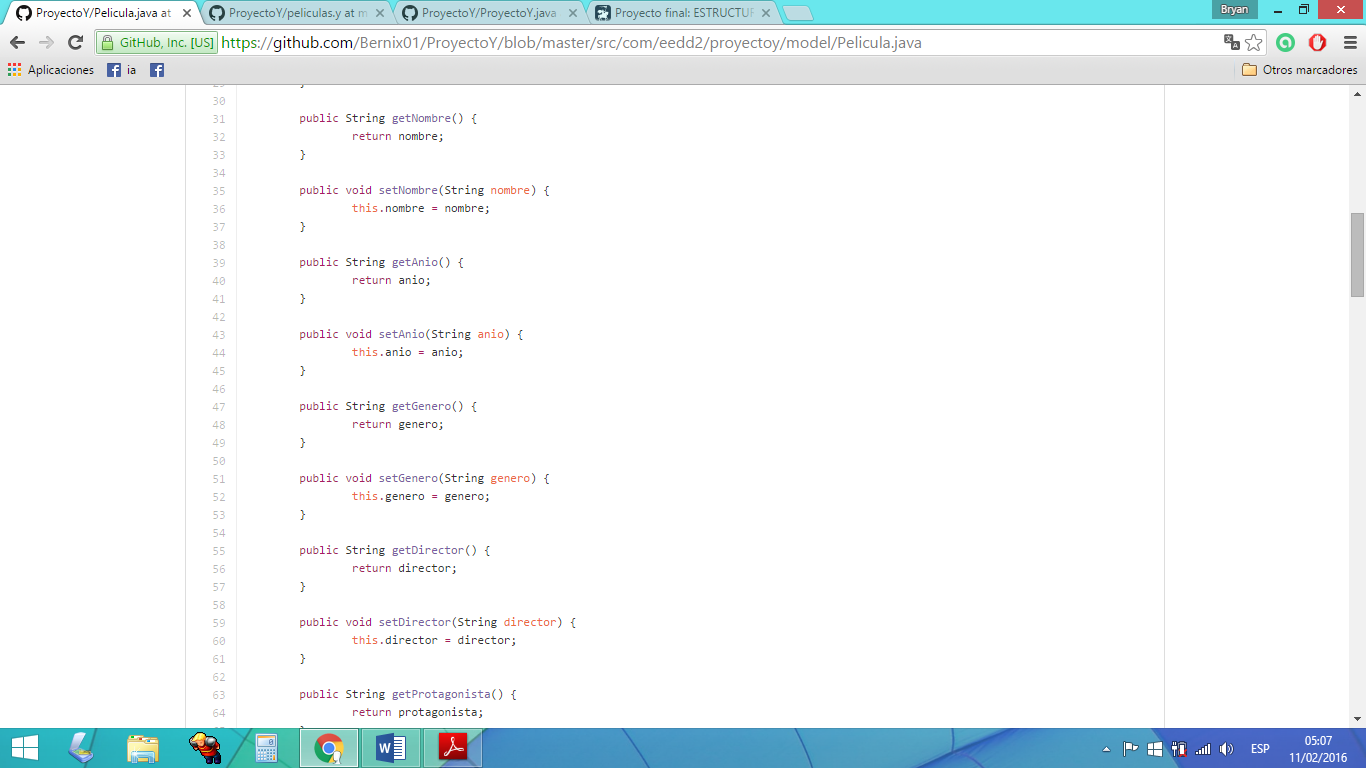
* VoteUp()

Este método realiza lo específicado en el proyecto de aumentar en 1 el rating de la película cada vez que esta es buscada.



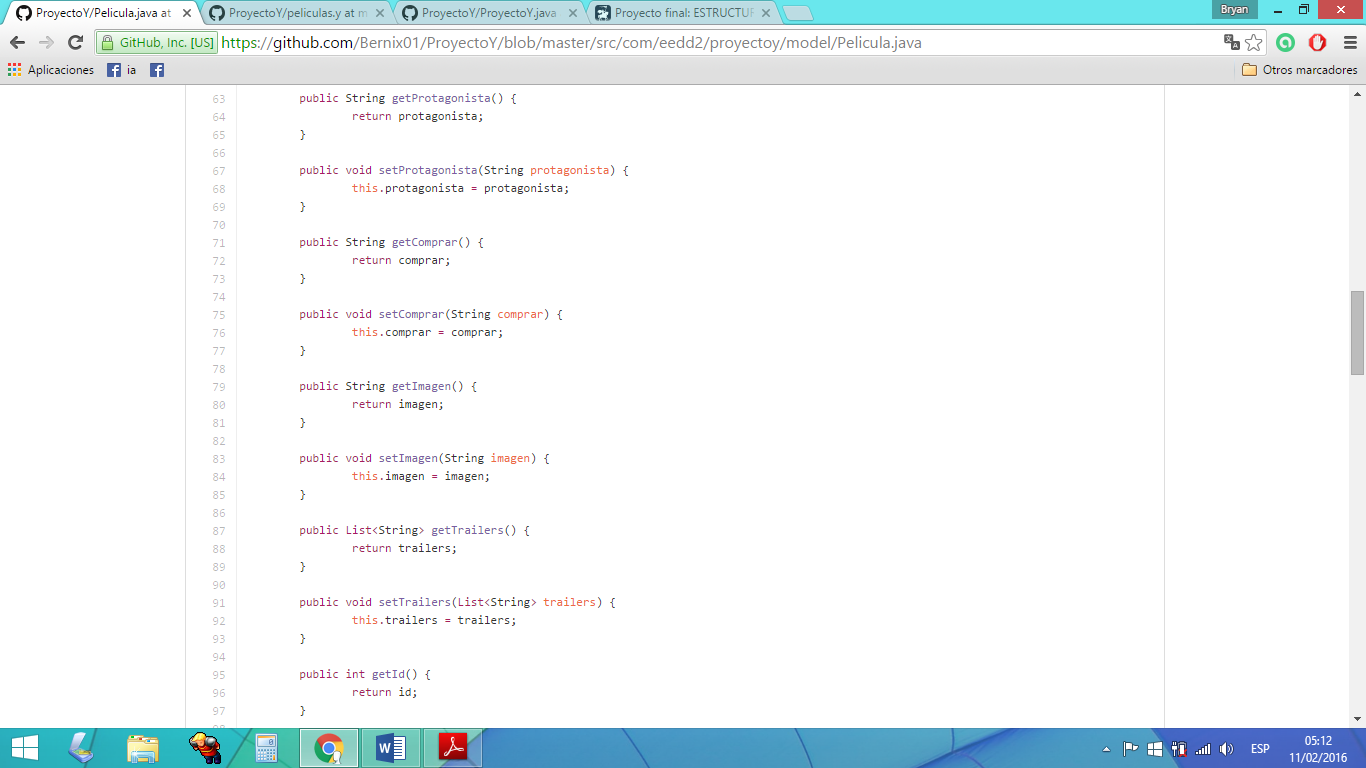
* Get/SetDirector()

El método Get permite obtener el dato privado Director del objeto Pelicula. El método Set permite cambiar el director del objeto Película.



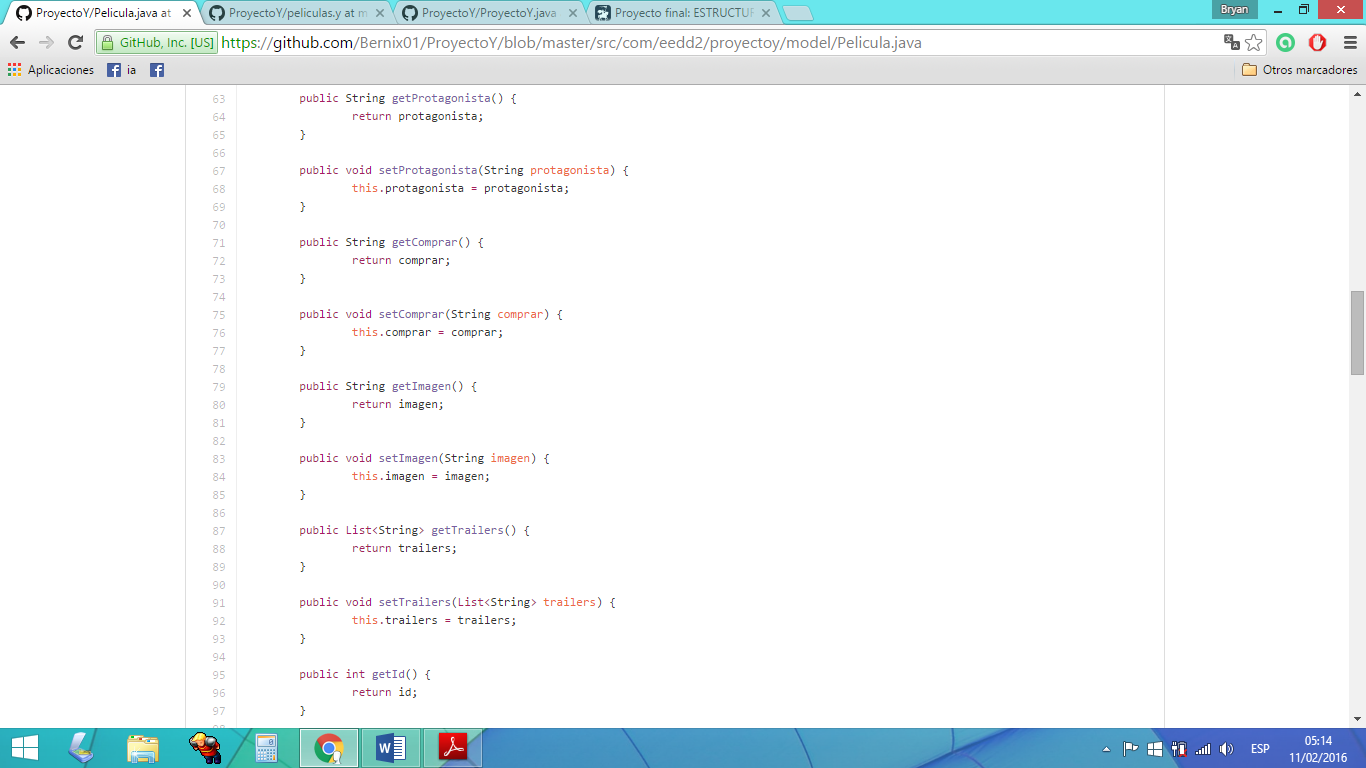
* Get/SetProtagonista()

El método Get permite obtener el dato privado Protagonista de Pelicula. El método Set permite cambiar el protagonista del objeto película.



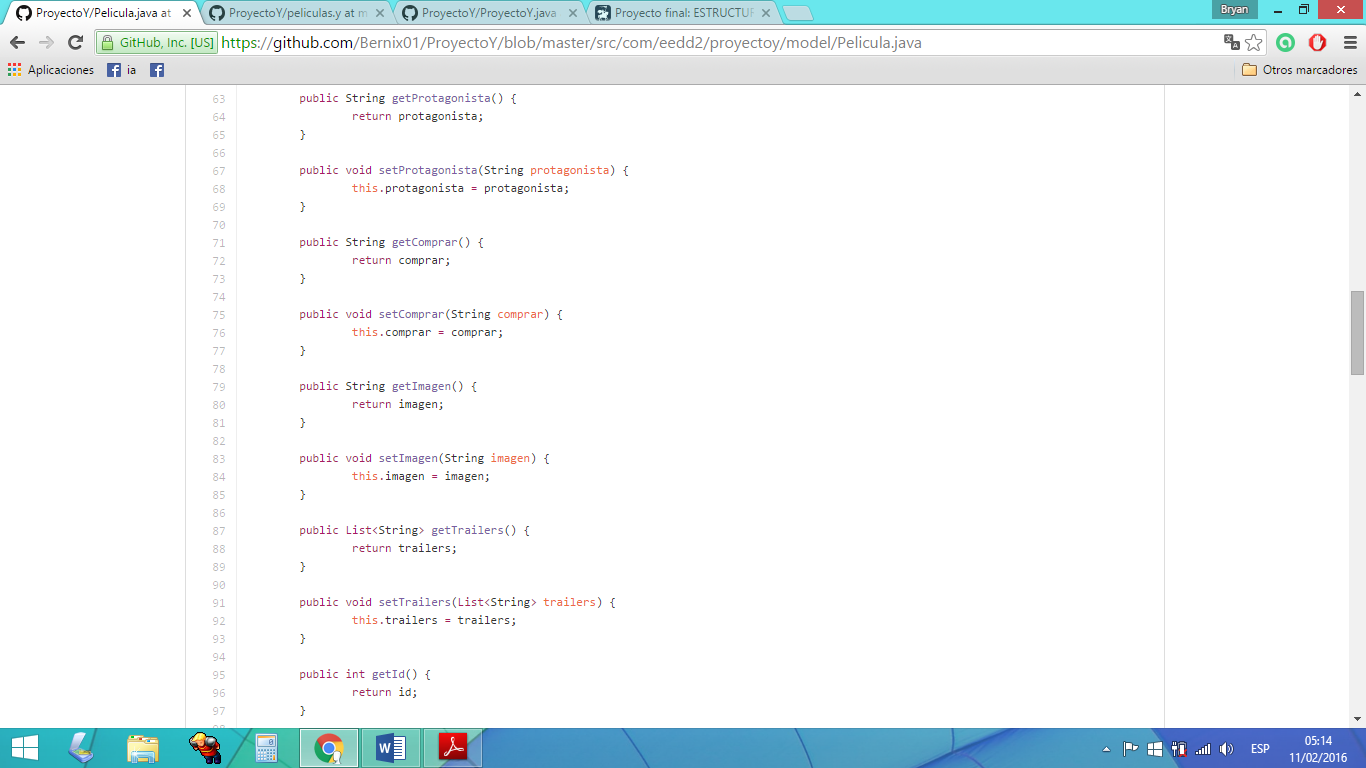
* Get/Set Comprar()

El método Get permite obtener el dato privado Comprar de Pelicula. El método Set permite cambiar el protagonista del objeto película. Se puede usar para cambiar el link de compra de la película.



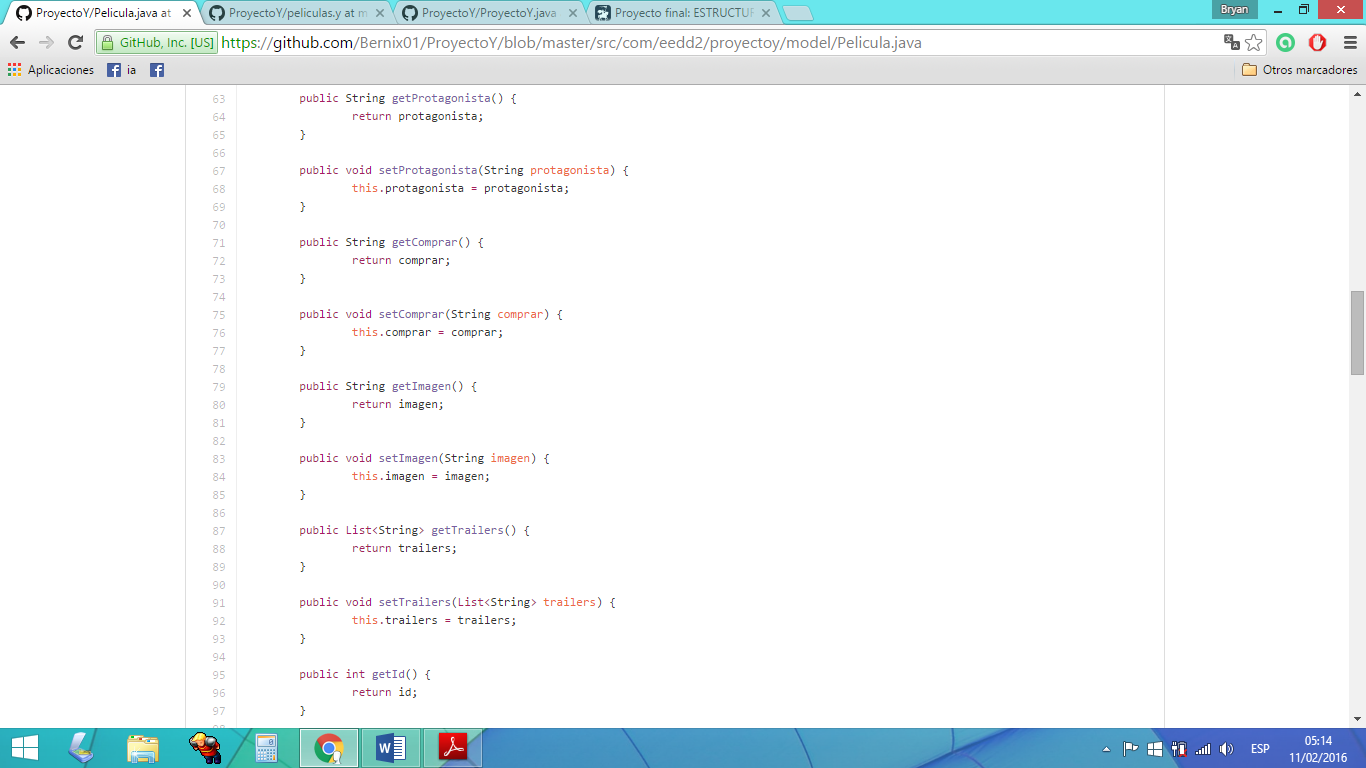
* Get/Set Trailer()

El método Get permite obtener el dato Trailer de la película. El método Set permite cambiar o definir el link donde se puede ver este tráiler.



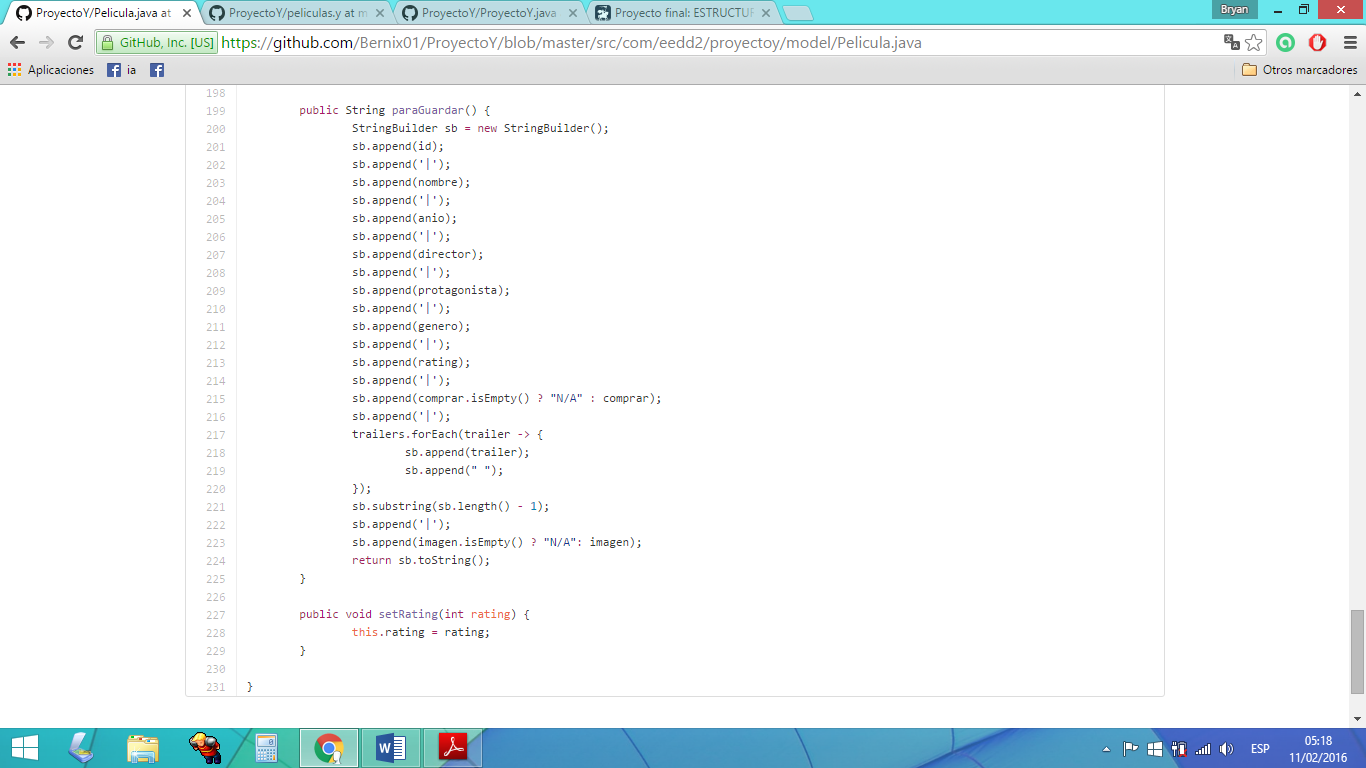
* Get/Set Imagen()

El método Get permite obtener el dato imagen de la película. El método Set permite cambiar o definir el link donde se puede ver esta imagen.

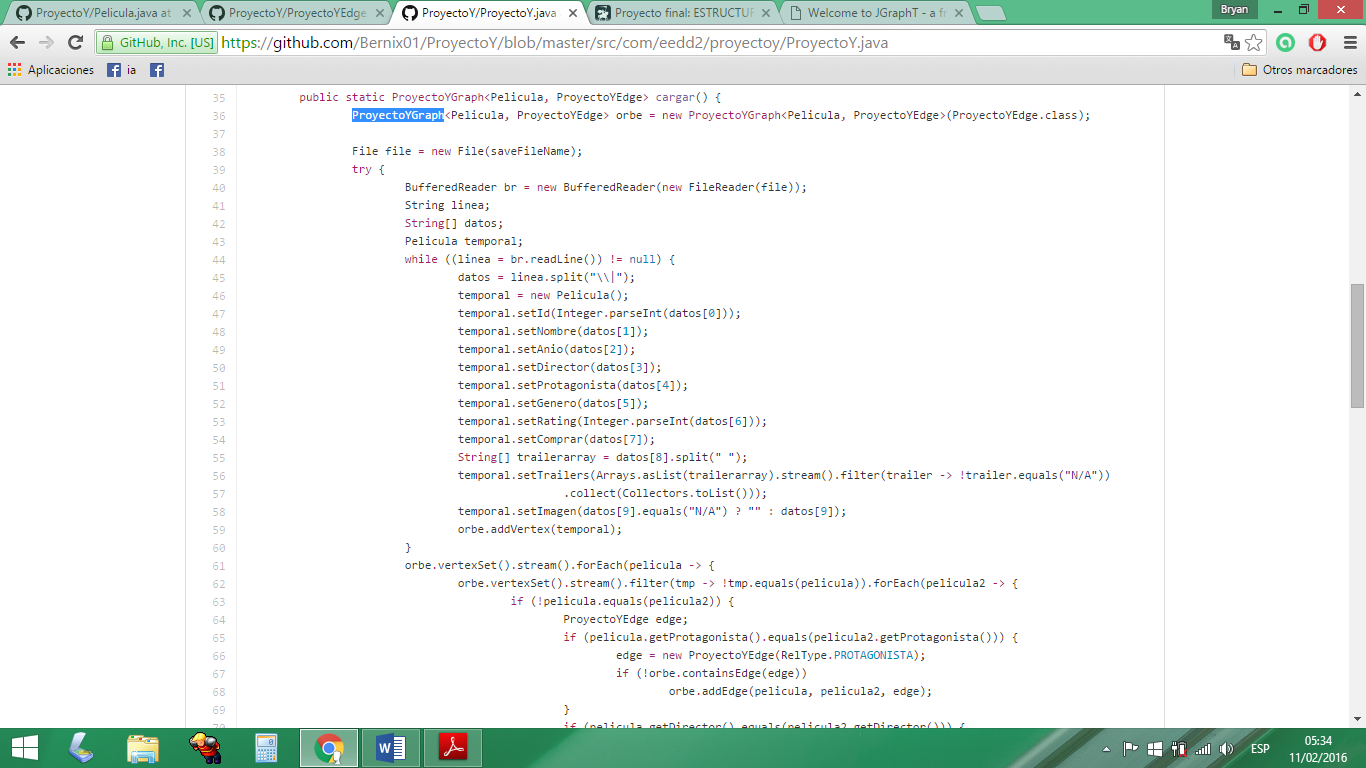


* paraGuardar()

Este método retorna un conjunto de texto donde se encuentra los datos id, nombre, anio, genero, director, protagonista, comprar, tráiler, imagen. Cada uno de estos datos se encuentran separados por el símbolo “|”. Esto es lo que se usará si se quiere ingresar alguna nueva película a la base de datos.



El TDA películas es el más importante pues contiene los datos en bruto de lo que será nuestra aplicación. Y el otro TDA de suma importancia, por el cual se podrá realizar el proyecto es el grafo ProyectoYGraph “orbe” el cual tendrá los datos de todas las películas y nos ayudará mucho en las relaciones tales como la búsqueda por nombres, relación de directores y por protragonistas.

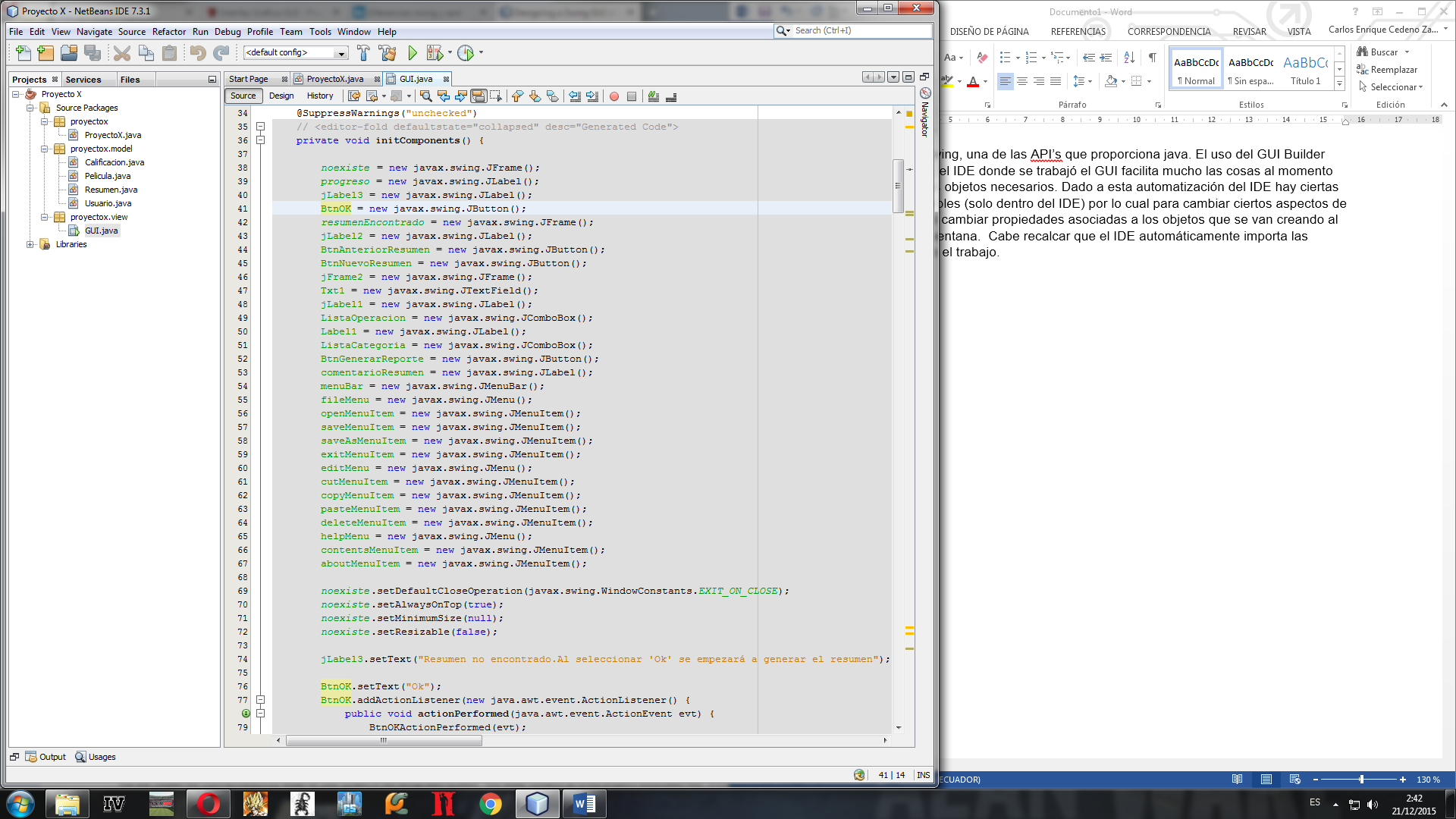


# **Mostrar y detallar los conceptos de la materia que se encuentran implementados en su código.**

* **Grafos:** Un grafo en el ámbito de las ciencias de la computación es un tipo abstracto de datos (TAD), que consiste en un conjunto de nodos (también llamados vértices) y un conjunto de arcos (aristas) que establecen relaciones entre los nodos **.**La principal complejidad que hay en este proyecto es la debida creación de los grafos, y su uso. Una vez dominado esto, el proyecto se vuele mucho más sencillo. Nuestro grupo obvió este problema gracias al uso de la librería JGrapht (<http://jgrapht.org/>) con la cual se pudo trabajar rápidamente. Dentro del proyecto, el grafo orbe que usamos es de vital importancia para la eficiencia a la hora de buscar comparaciones. Uno de los requerimientos dentro del proyecto es que se pueda buscar por directores y por protagonistas lo cual se hace recorriendo el grafo por anchura, buscando el requerimiento que solicitado por el usuario.
* **Interfaz Gráfica GUI (Graphical User Interface)** es el conjunto de componentes gráficos que posibilitan la interacción entre el usuario y la aplicación. Es decir ventas, botones, listas, cajas de diálogo, campos de texto, etc.

Para nuestro proyecto usa Swing, una de las API’s que proporciona java. El uso del GUI Builder proporcionado por Netbeans, el IDE donde se trabajó el GUI facilita mucho las cosas al momento de declaras las variables y los objetos necesarios. Dado a esta automatización del IDE hay ciertas partes del código no modificables (solo dentro del IDE) por lo cual para cambiar ciertos aspectos de la programación es necesario cambiar propiedades asociadas a los objetos que se van creando al momento de ubicarlos en la ventana. Cabe recalcar que el IDE automáticamente importa las librerías necesarias para todo el trabajo.

El método con que Netbeans inicializa los componentes de nuestro GUI es initComponents(), en donde al momento de inicializar los componentes asigna los valores por defecto. A medida que se vayan modificando estos valores en la ventana de diseño el código automáticamente se modifica. Para nuestro GUI, el código generado en este método consta de alrededor de 300 líneas.



Dentro de los métodos actionPerformed() correspondientes a cada componente de nuestro entorno se establecen líneas de código a realizarse en ‘caso de’.

# **Explicar errores que se tuvieron durante la implementación del codio mediante imágenes de pantalla capturadas que muestren operaciones de depuración (bug) en los cuales se visualicen las variables y sus valores. Explique cuál era el error y cuál fue su solución.**

**ERRORES GENERALES (CÓDIGO JAVA).**

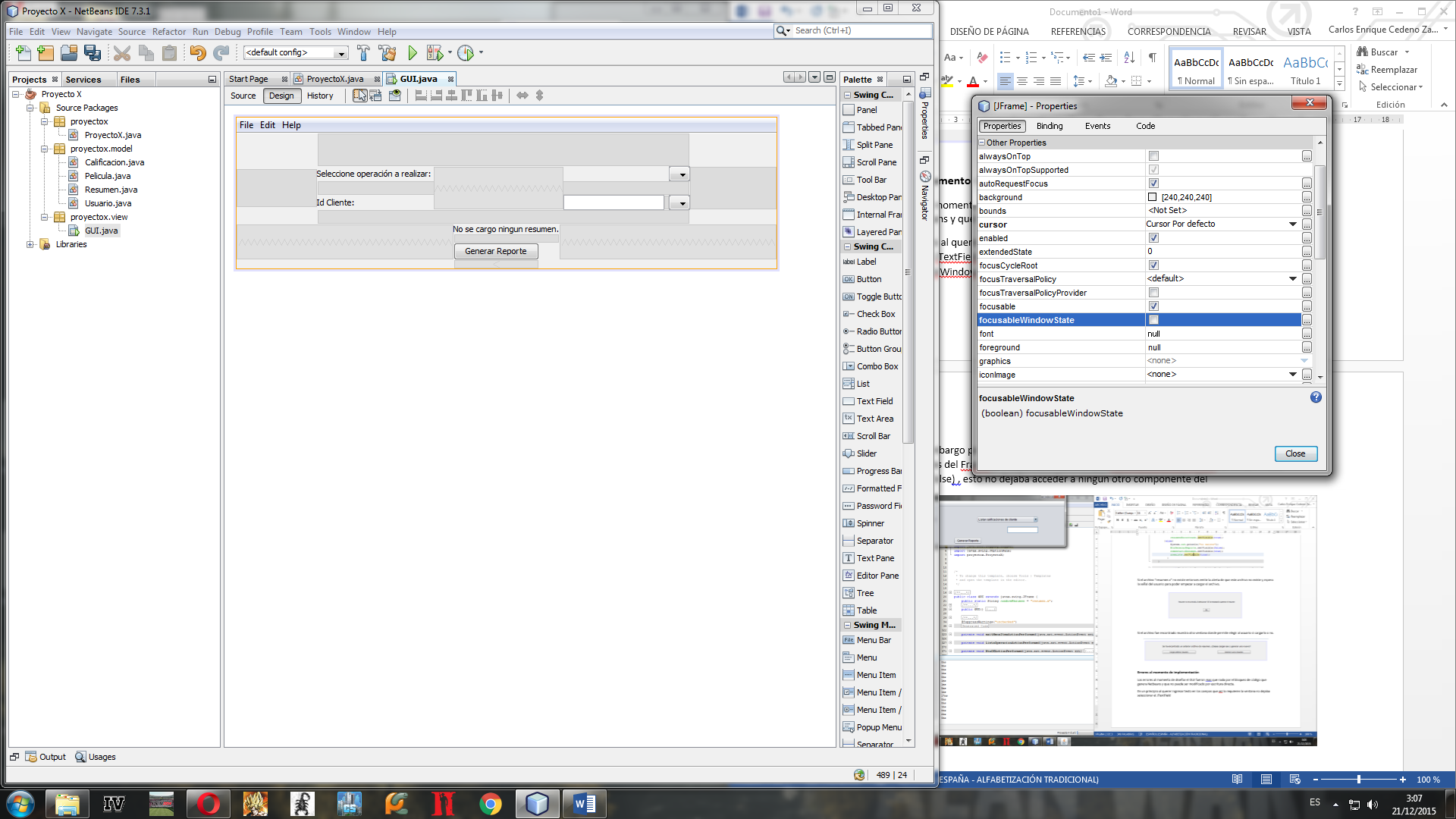
Uno de los problemas más frecuentes para mí, Bryan Ordóñez, por lo cual no pongo siquiera impresiones de pantalla fueron los errores de sintaxis. Debido a que en el semestre anterior usé Python, mi lenguaje de programación está grandemente influido, y cuando programaba con apuro, no podía evitar poner varias líneas como si este fuera Python. Debido a esto, mis problemas fueron:

* Errores de sintaxis: Es el error más común al mesclar lenguajes de programación, por ejemplo los bucle for que usaba, tenían expresiones de Python.
* Errores de Casting

**ERRORES AL MOMENTO DE DISEÑAR EN GUI**

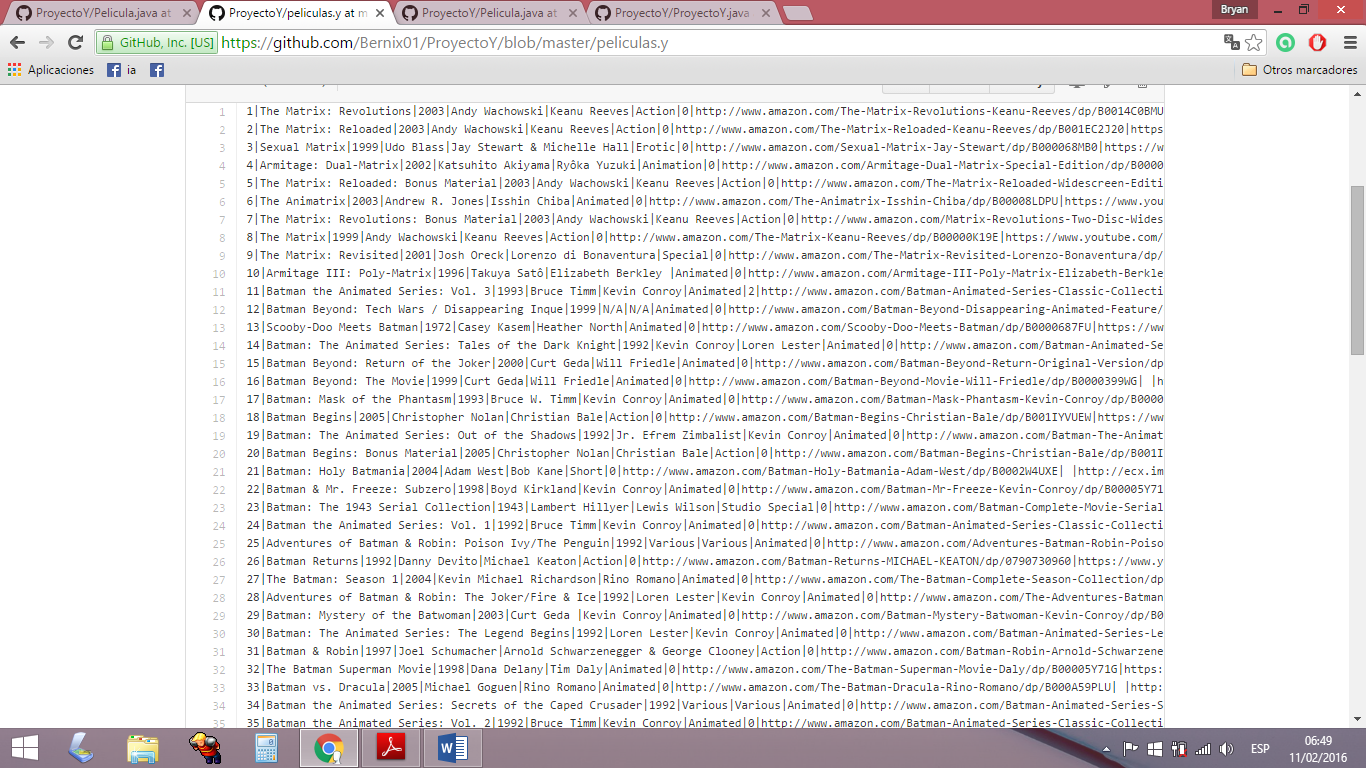
Los errores al momento de diseñar el GUI fueron más que nada por el bloqueo de código que genera Netbeans y que no puede ser modificado por escritura directa.

En un principio al querer ingresar texto en los campos que asi lo requieren la ventana no dejaba seleccionar el JTextField por ningún método. Eran aplicados focusRequest() y requestFocusInWindow() los cuales no necesariamente pueden ir en el código, solo facilitan el uso del GUI, sin embargo por ningún método era ‘alcanzable’ el campo de texto. Hasta que revisando las propiedades del Frame principal se encontró que FocusableWindowState se encontraba desactivado (false), esto no dejaba acceder a ningún otro componente del Frame



Esto pudo ser solucionado activando esa opción. Otro error es que por defecto los frames creados no incluyen el método pack(), el cual le otorga a los componentes su tamaño preferido establecido.

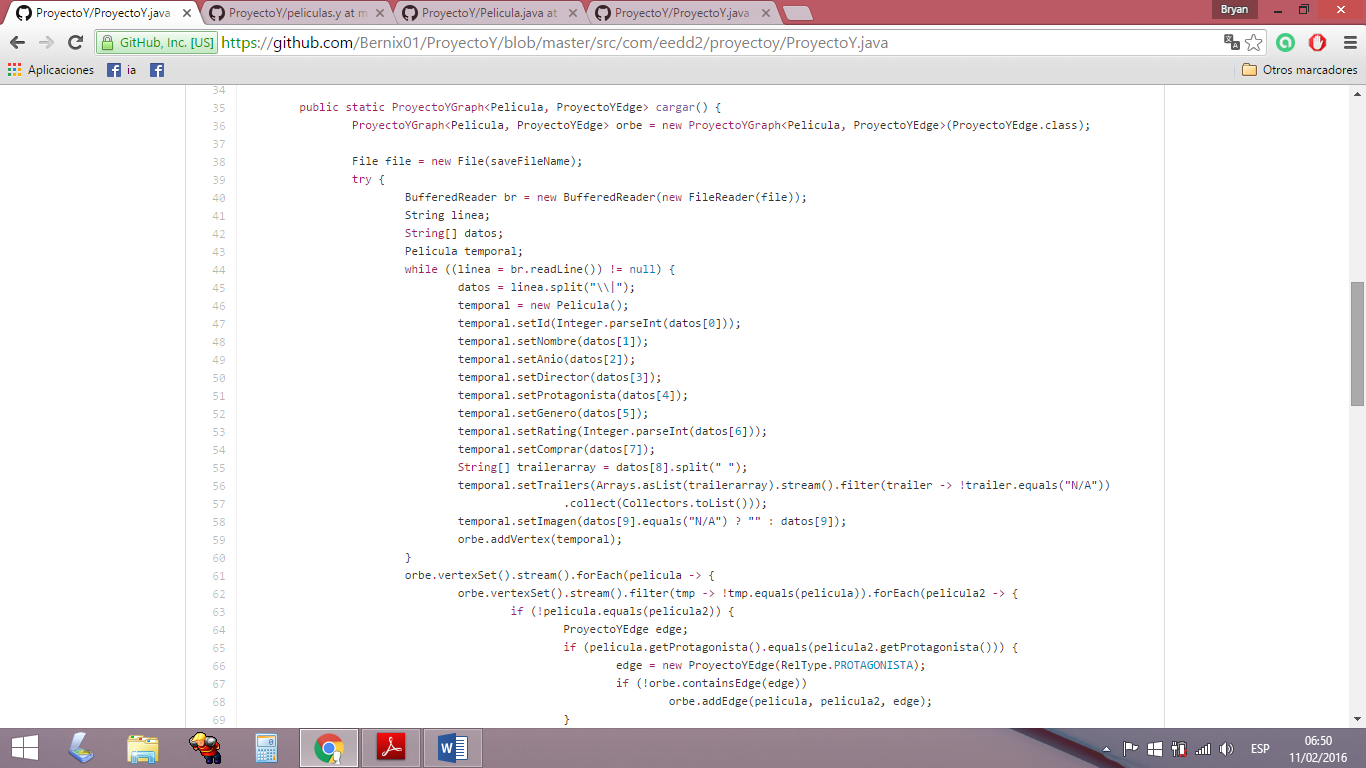
**OTROS ERRORES**



**Problema:** Algunas películas no contenían el rating

**Error:** Integer.parseInt() no puede procesar “NULL”.

**Solución:** agregar condicional para “NULL” e inicializar con 0;



**Problema:** datos resumidos no cargaba.

**Error:** Carácter “|” no escapado correctamente.

**Solución:** cambiar el parámetro de linea.split(“|”) a línea.split(“\\|”).

# **Describir el uso de su aplicación, es decir ejecútelo como usuario del mismo y explique cada una de las operaciones siguiendo un flujo de ejecución.**

1. Al ejecutar la aplicación se carga el archivo de un grafo y se establecen las relaciones de semejanza con respecto al director y protagonista.
2. Luego se solicita al usuario ingresar una palabra con la cual buscar.
3. Una vez ingresada la palabra, se muestran todas las posibles coincidencias que contengan esa palabra en el nombre.
4. Cuando el usuario selecciona una película, aumenta en uno el rating de la película, se sincroniza el grafo con el archivo.
5. Se muestra el html con los resultados y sus detalles.

