

Sergio Cobos García

Yura Turlay

David Luna Fernández

Proyecto CFGS 2º Desarrollo de Aplicaciones Web

**Índice**

Introducción

Tecnologías usadas

Diseño de la aplicación web

Diseño de la Base de datos

Desarrollo y despliegue

Bibliografía

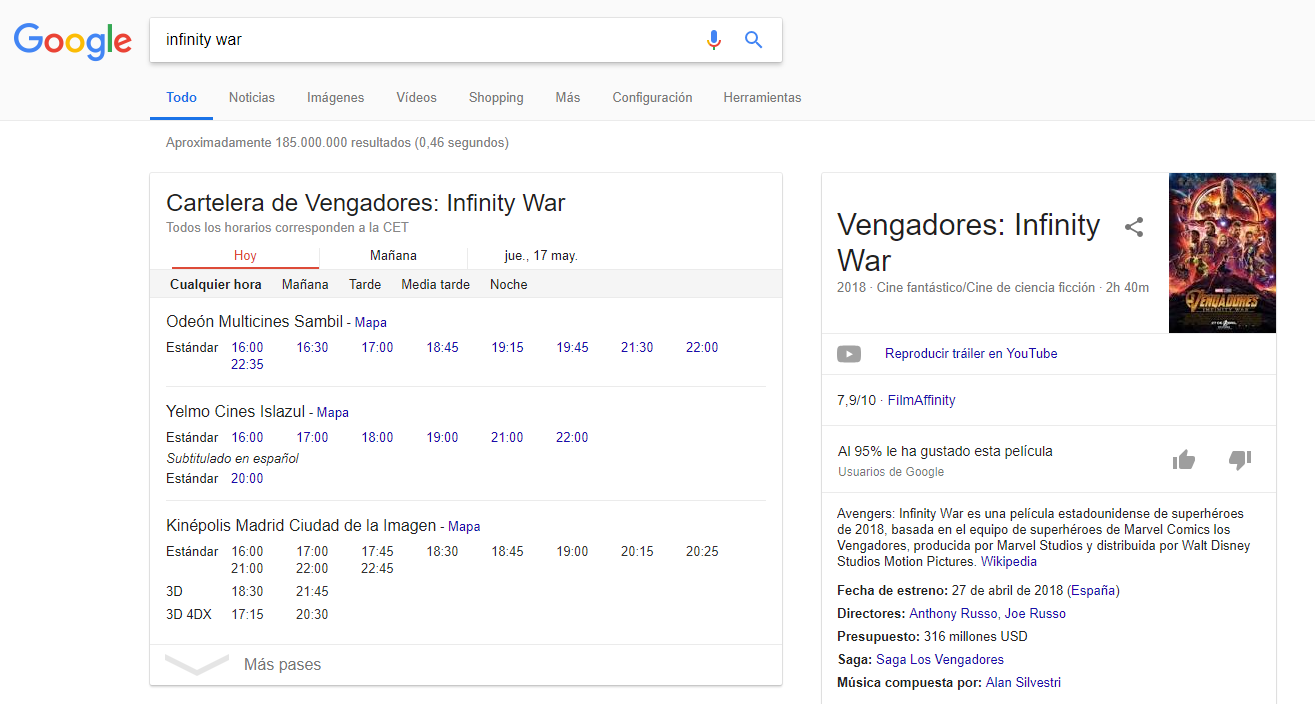
**Introducción**

YoSoyTuCine es un buscador/comparador de cines, no de películas.

Entender esta idea es básica para comprender el proyecto que abordamos ya que comparadores y buscadores de películas hay muchos e imposible de competir con ellos debido a su gran comunidad de usuarios, véase FilmAffinity.

La idea del proyecto nace de la idea de poder tener en simple vistazo, gracias a la geolocalización, tu cine más cercano.

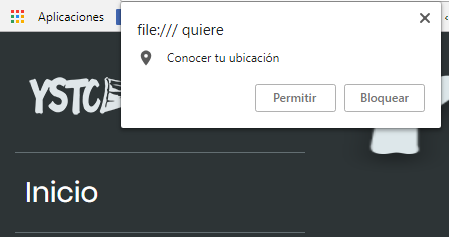
Por ejemplo, Google, a fecha de entrega de este proyecto, si buscas una película, la información que te propone son las sesiones y cines disponibles, pero no tu cine más cercano.



La idea de YoSoyTuCine es además ordenar las películas primeramente por cercanía a tu ubicación en todo momento, y con posibilidad a hacerlo en base a las valoraciones dadas a los cines por la comunidad de usuarios de YoSoyTuCine.

Otro punto a destacar de YoSoyTuCine es que evidentemente al no poder competir con buscadores omnipotente como Google, intentamos ofrecer la información buscada por el usuario en un menor número de clic, esa es la ventaja que proponemos, ya que simplemente al entrar en la web, serás preguntado por tu ubicación, tendrás ante tus ojos lo carteles de las películas en cartelera actualmente y simplemente pinchando en el cartel tendrás todos los cines a tu alrededor ordenados por distancia y la información de las sesiones, cines y película en la que acabas de hacer clic.

Para lograr los objetivos finales de YoSoyTuCine hemos de usar las funciones de geolocalización de Google para poder establecer nuestra ubicación,





y poder hacer los cálculos de distancia por latitud y longitud entre nuestra ubicación actual y la del cine, la cual es almacenada en base de datos y que es rellenada automáticamente tanto en cines como en películas que están en cartelera mediante la técnica de scrapping.

El scrapping es básicamente obtener información de una web, aquí vamos a ver las definiciones:

*Web scraping es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en la World Wide Web ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.*

*El web scraping está muy relacionado con la indexación de la web, la cual indexa la información de la web utilizando un robot y es una técnica universal adoptada por la mayoría de los motores de búsqueda. Sin embargo, el web scraping se enfoca más en la transformación de datos sin estructura en la web (como el formato HTML) en datos estructurados que pueden ser almacenados y analizados en una base de datos central, en una hoja de cálculo o en alguna otra fuente de almacenamiento. Alguno de los usos del web scraping son la comparación de precios en tiendas, la monitorización de datos relacionados con el clima de cierta región, la detección de cambios en sitios webs y la integración de datos en sitios webs*

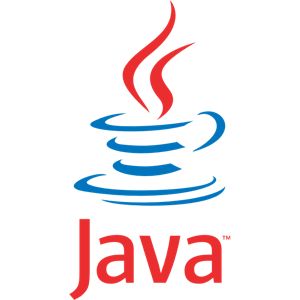
En nuestro caso el uso del scrapping es para obtener de las paginas como Cinesa por ejemplo, la información de una película y poder rellenar nuestra base de datos con la información obtenida, por ejemplo lanzando el scrapping cada viernes podemos actualizar nuestra base de datos con las películas nuevas o borrar las películas que ya no tenemos en cartelera.

Aquí podemos ver un ejemplo de código de scrapping donde recuperamos información de una película de la web de Cinesa:

Para poder tener el feedback de los usuarios lo que hemos decidido es usar la API de google para así evitar que los usuarios deban registrarse de cero en nuestra página, ya que es algo que cada vez menos le apetece realizar a la gente.

De esto modo con tu usuario de Twitter y mediante el uso de #Hashtags puedes interactuar con YoSoyTuCine.

**Tecnologías usadas**

****

****

****

****

****

****

**JavaScript**

Utilizado para generar el dinamismo de la web en conjunción con JQuery.

**Ajax**

Utilizado para evitar las recargas de la web a la hora de filtrar en las búsquedas.

**Java**

Utilizado bajo el patrón MVC encargado de dirigir la aplicación web, recibir peticiones, ir a base de datos y devolver vistas.

**MySQL**

MySQL ha sido el gestor de base de datos elegido debido a su facilidad de uso y a su ligereza que hacen de él la opción más utilizada para este tipo de proyectos Web.

**Bootstrap**

Librería de estilos creada y liberada por Twitter con una gran popularidad y aceptación en el mundo del diseño web gracias a las facilidades que aporta y que usamos en el proyecto como es su sistema de rejilla.

**HTML5**

Estándar, es el esqueleto de la aplicación web.

**CSS3**

Estándar en cuanto a estilos de una web.

**JQuery**

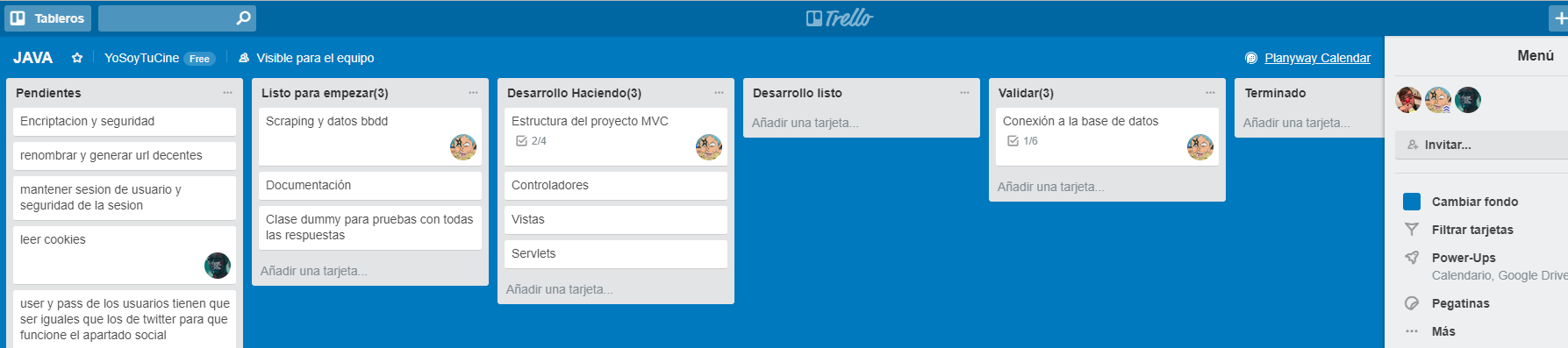
Encargado de generar HTML dinámicamente y elegido por su rapidez y sencillez en cuanto a código para realizar dicha tarea.

Todas estas tecnologías han sido abordadas en mayor o menor medida durante el ciclo y por tanto no es necesario desarrollarlas más en esta documentación, a continuación explicaremos más detenidamente dos de las cuales no se abordaron durante el curso.

**Trello**

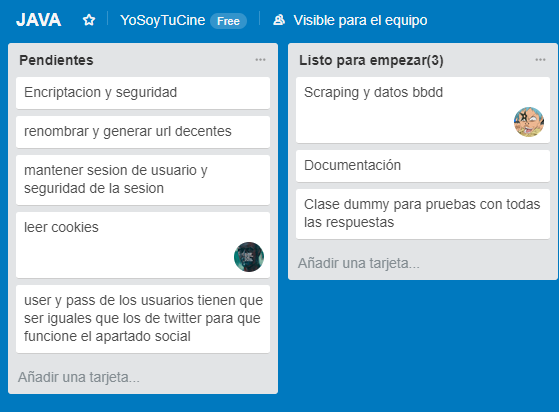
Trello es una aplicación Web la cual te proporciona un sistema de trabajo basado en metodologías agiles.

Lo que podemos ver aquí es un tablero Kanban en el cual se distribuyen las tareas entre los distintos miembros del equipo y durante las distintas fases del desarrollo de una tarea



En nuestro caso hemos creado distintos tableros para diferenciar tecnologías, por ejemplo aquí vemos el tablero para las tareas que tienen que ver con JAVA.

A su vez disponemos de otros tableros creados para las tareas de Base de Datos, de HTML, CSS y JS… etc.



Los estados por los cuales pasa una tarea son los vistos en las capturas adjuntas:

Pendiente: Donde se ponen las tareas que necesitaremos, es una pila de las tareas que englobaran el proyecto.

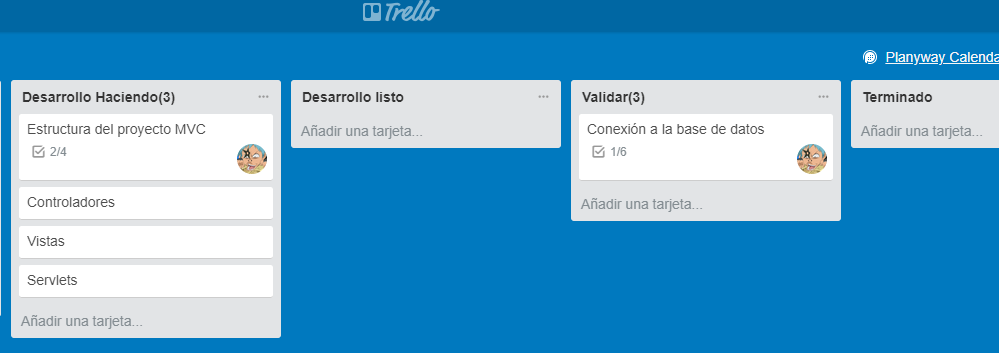
Listo para empezar: significa que dicha tarea ya tiene las dependencias necesarias completadas y está lista para ser desarrollada.

Haciendo: Tarea en desarrollo.

Listo: Es donde se colocan las tareas terminadas listas para que el encargado del equipo de la validación coja la tarea y se la lleve a su campo y pueda revisarla y decida si tiene que volver a desarrollo o está terminada

Validar: Tarea en validación.

Terminado: Utilizado como guía visual del trabajo realizado hasta ahora



Cada tarea puede tener subtareas, adjuntos, tags, fechas de entrega…



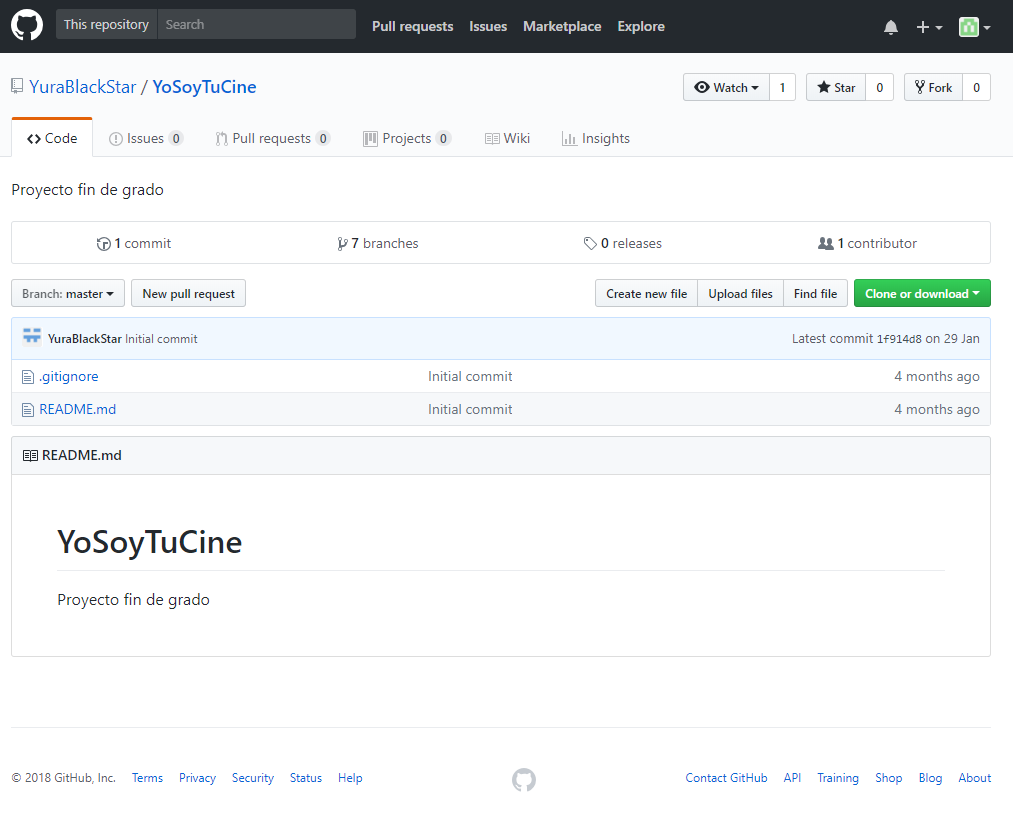
Como vemos Trello es una aplicación de gran utilidad para controlar y organizar el trabajo de un equipo de desarrollo, y es ampliamente usado en todo el mundo del desarrollo.

**Github**

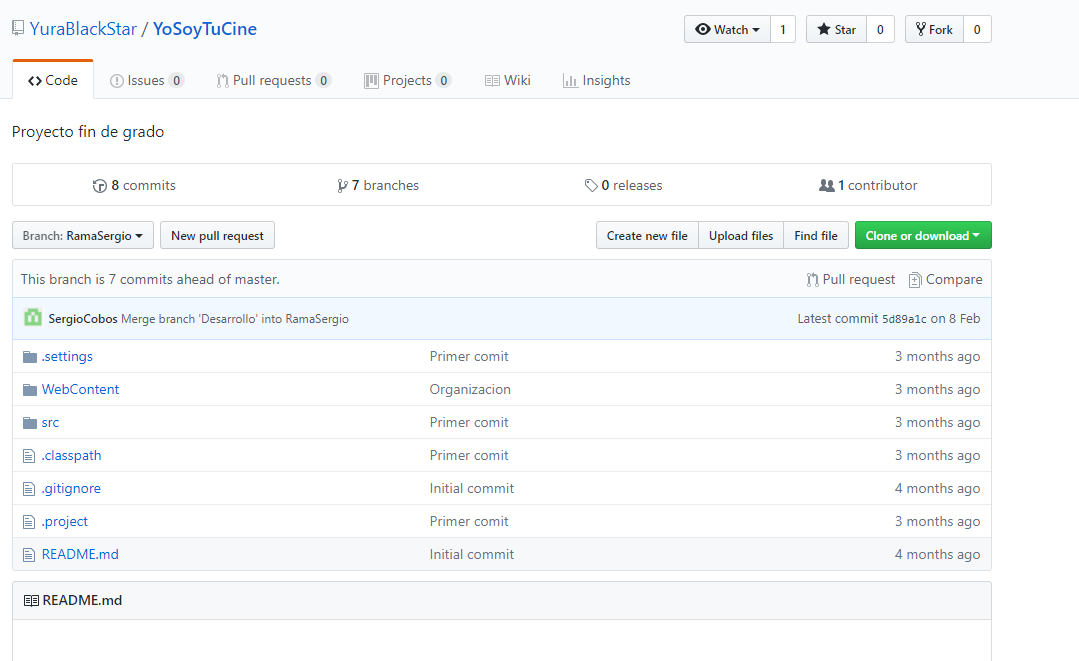
Github es la comunidad/red social de los programadores por excelencia y está basado en el software de control de versiones GIT, creado por Linus Torvalds.

Para nuestro proyecto hemos decido usar control de versiones para familiarizarnos con esta herramienta absolutamente imprescindible y fundamental en el mundo del desarrollo.

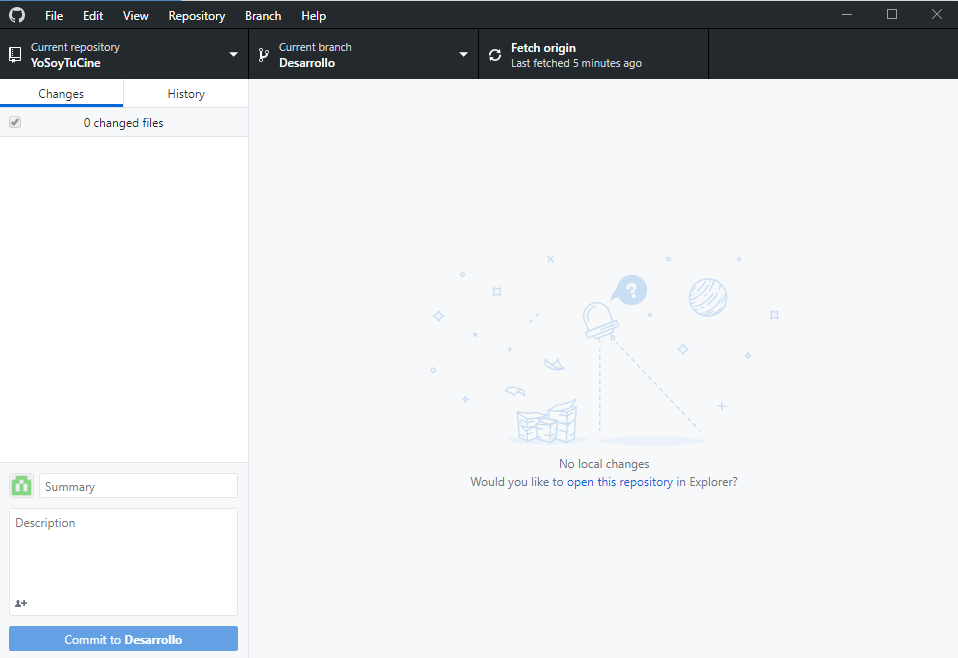
Creamos un repositorio para el proyecto en el cual los integrantes del equipo de desarrollo podemos programar simultáneamente al disponer siempre de las versiones actualizadas de los códigos y gracias a funcionalidades de los software de control de versiones como es el Merge que te permite que varias persones trabajen sobre un mismo código integrando al finalizar de programar las líneas de todo el equipo.



La forma elegida para trabajar ha sido la de crear una rama master, una rama desarrollo y varias ramas individuales para trabajar cada uno sobre nuestra rama e ir integrándolas sobre la rama desarrolla y una vez esta testeado integrarlo en la master, el control de versiones te permite tener un absoluto control sobre las diferentes versiones del software y te permite volver atrás y deshacer errores con gran facilidad y mantener un orden sin volverte loco



Puedes trabajar mediante la consola de GIT y tener acceso a todas los comandos o para familiarizarte y aprender las funciones más básicas (en nuestro caso no tenemos que hacer uso de comandos avanzados) usar el cliente de escritorio, que es tremendamente sencillo y fácil de usar, a simple vista podemos ver cómo podemos elegir las ramas, descargarnos la última versión del código, hacer commit de los cambios, hacer merge de ramas, subir los cambios al repositorio remoto…etc.



**Diseño de la web**

**(DESARROLLAR ESTE PUNTO)**

Lo primero que tuvimos que realizar fue un análisis de requerimientos

Primero hemos diseñado los bocetos con herramientas de prototipado tales como… generando los siguientes diseños

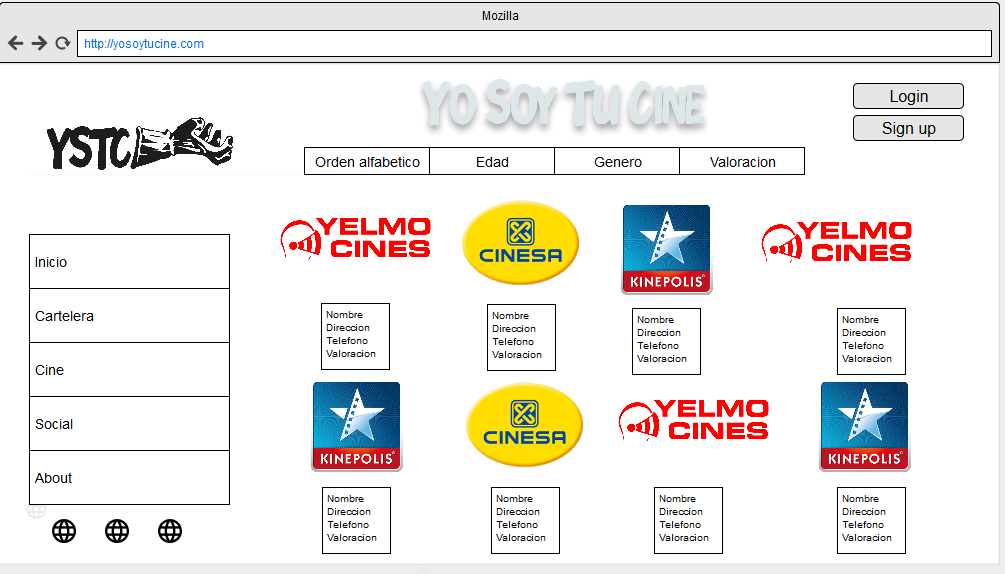






Finalmente fuimos creando la estructura html y los css

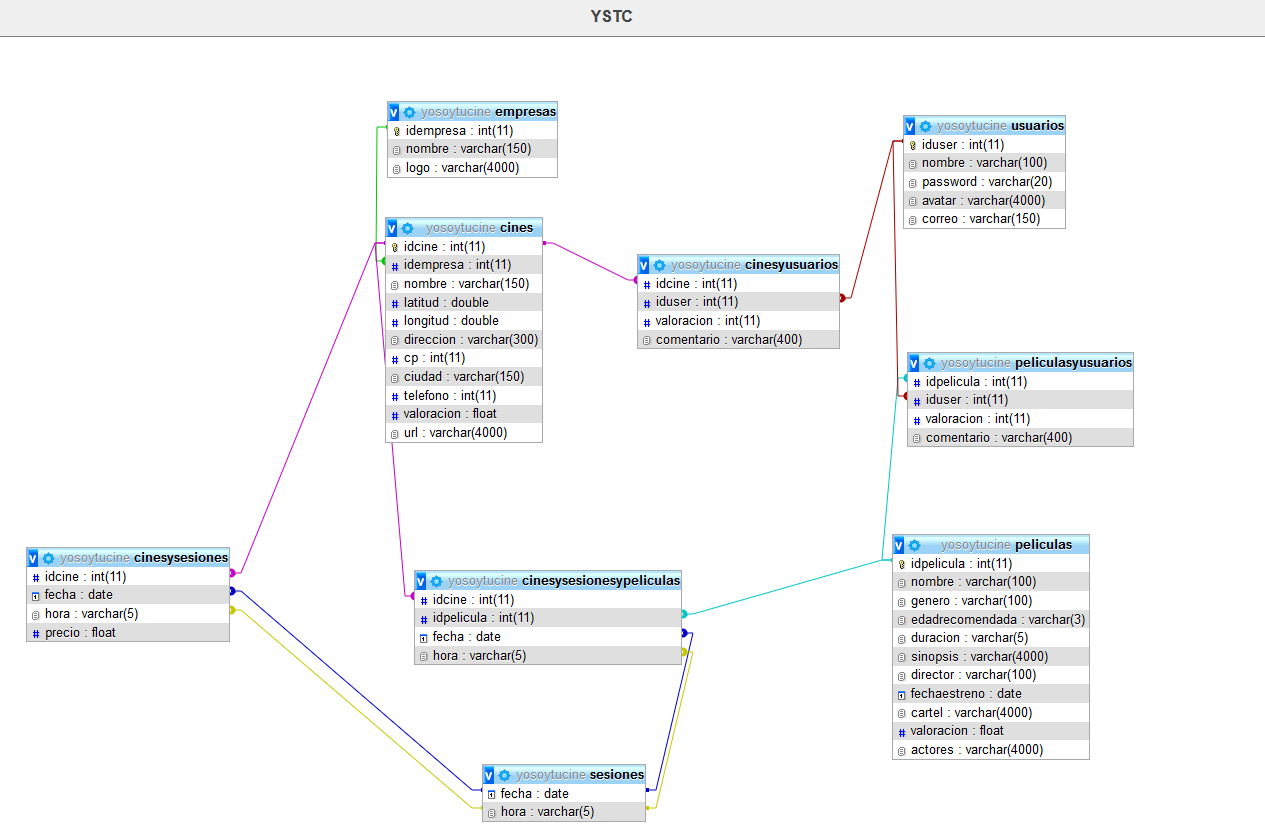




**Diseño de la base de datos**

**(DESARROLLAR ESTE PUNTO)**

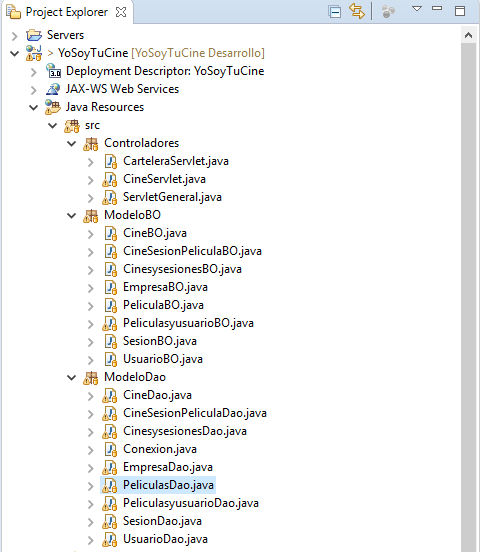
Se realiza el modelo entidad relación, se pasa a modelo relación y finalmente se crea físicamente con los siguientes scripts.

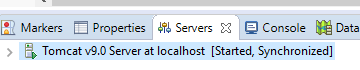
Diagrama

**Desarrollo y despliegue de la aplicación**

**(DESARROLLAR ESTE PUNTO)**

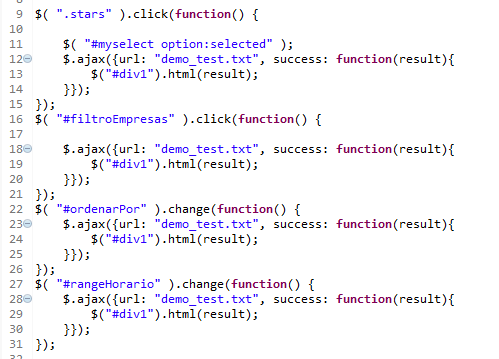
La aplicación está realizada bajo el patrón de diseño Modelo-Vista-Controlador (MVC), para su codificación hemos utilizado eclipse que se integra perfectamente con GIT y hemos utilizado tomcat para desplegar la aplicación.





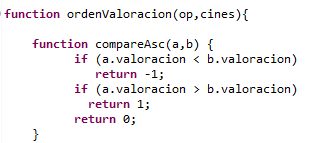
Aquí podemos ver algunas capturas con código interesante nuestra aplicación

Llamadas Ajax

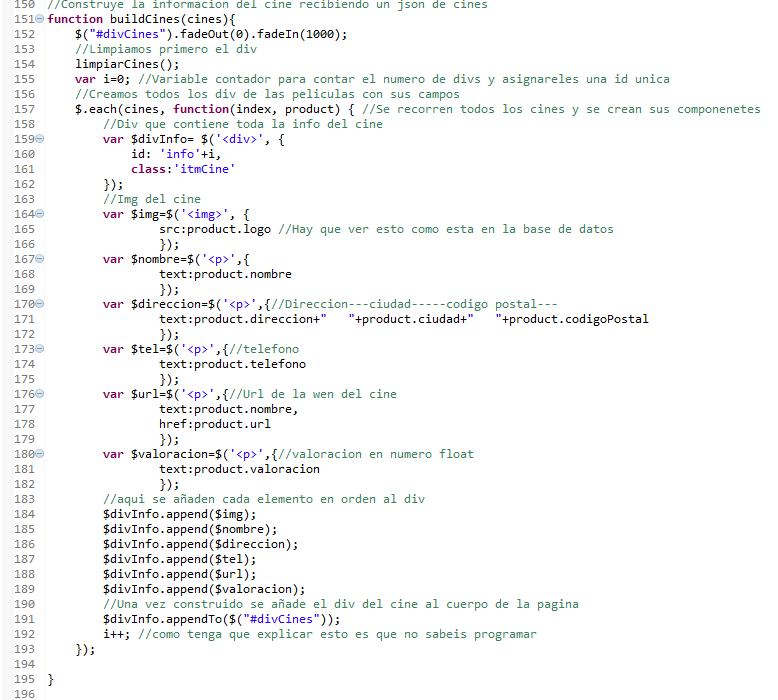


Métodos Java en los filtros de las web





Generación de HTML mediante JQuery



Captura de la web funcionando

**Bibliografía**

**(DESARROLLAR ESTE PUNTO)**