

Showganizer

EQUIPO:

Pedro Pablo Aste Kompen – Katherine Páez Ramos

Resumen de Actores

Nombre	Descripción
Usuario Administrador	Es parte de la comunidad de la plataforma, pero tiene permisos avanzados que le permiten agregar/borrar contenido de la web como series, películas, usuarios, posts y reviews. Mantiene la plataforma web en orden.
Usuario registrado	Es parte de la comunidad de la plataforma. Tiene su cuenta propia por lo que accede a todas las funcionalidades disponibles. No solo usa la web para mantener una lista de sus películas y series, sino que comparte información a través de publicaciones. Visita varias veces por semana la web.
Visitante	Usa la plataforma, pero solo le interesan los datos sobre las series y películas, sus sesiones son rápidas y puntuales.

Casos de uso

CDU01:

Nombre:	Inscripción de Usuario
Descripción:	El usuario hace su registro completo en el sistema ingresando sus datos personales y al terminal recibe una credencial válida.
Actores:	Visitante
Pre-condiciones:	Ninguna
Post-condiciones:	El usuario queda registrado en la base de datos. Además, queda un registro en el log.
Gatilladores:	El usuario solicita registrarse.

Flujo Básico:

1. Se abre el formulario de nuevo usuario.
2. Usuario llena sus datos.
3. Usuario presiona en Enviar para terminar su registro.
4. Se guardan los datos en la base de datos.

5. Se muestra un mensaje de "Registro Exitoso"

Flujo Alternativo:

El usuario no ingresó un campo obligatorio

1. Aparecerá un mensaje de error y se destacarán las casillas que faltan.

Formato de e-mail inválido

1. Aparecerá un mensaje de error y se pedirá reingresar el mail.

E-Mail ya registrado

1. Aparecerá un mensaje de error y da la opción de ir a la ventana de recuperación de clave.

Falla conexión a Base de Datos

1. Aparecerá un mensaje de error y se dará la opción de reintentar el ingreso.

CDU02:

Nombre:	Identificación de Usuario
Descripción:	Un usuario se identifica a través del ingreso de su e-mail y contraseña.
Actores:	Usuario Registrado
Pre-condiciones:	El usuario debe estar inscrito previamente en el sistema
Post-condiciones:	Se deja un registro en el log
Gatilladores:	El usuario selecciona la opción para identificarse.

Flujo Básico:

1. Se muestra el formulario de identificación.
2. Usuario ingresa su e-mail y contraseña.
3. Usuario presiona Aceptar.
4. Se validan los datos y se redirige a la página personal.

Flujo Alternativo:

E-mail no está registrado en la base de datos

1. Aparecerá un mensaje de error pidiendo reingresar los datos o registrarse.
2. Se proveen las dos opciones, el usuario elige una.

Formato de e-mail no válido

1. Aparecerá un mensaje de error y se pedirá reingresar el mail.

No coinciden el mail y la contraseña

1. Aparecerá un mensaje de error y se dará la opción de reingresar la contraseña o ir a la página de recuperación de clave.

Falla conexión a Base de Datos

1. Aparecerá un mensaje de error y se dará la opción de reintentar.

CDU03:

Nombre:	Búsqueda de series y películas
Descripción:	El usuario hace una consulta por un cierto título utilizando las opciones de búsqueda habilitadas en el sitio.
Actores:	Usuario registrado, invitado
Pre-condiciones:	ninguna
Post-condiciones:	Aparecen los resultados del título buscado
Gatilladores:	El usuario selecciona el botón buscar

Flujo Básico:

1. El usuario escribe el título de algún material audiovisual.
2. El usuario hace *click* en buscar.
3. El usuario puede ver información de los resultados al hacer *click* sobre ellos.

Flujo Alternativo:

El usuario hizo su búsqueda y no hay resultados.

1. Se da la opción al usuario de volver a buscar.

CDU04:

Nombre:	Ver información sobre película, serie, temporada o episodio
Descripción:	El usuario al encontrar ítem (película, serie, temporada o episodio) en una búsqueda puede solicitar información adicional.
Actores:	Usuario registrado, invitado
Pre-condiciones:	Que el usuario haya finalizado con éxito una búsqueda.
Post-condiciones:	El usuario puede agregar comentarios al ítem y puntuar su calidad
Gatilladores:	El usuario hace <i>click</i> sobre un ítem que haya encontrado mediante una búsqueda.

Flujo Básico:

1. El usuario hace *click* sobre un ítem que haya encontrado mediante una búsqueda.
2. Se carga la sección donde se puede ver información del ítem.
3. El usuario puede ver comentarios que hayan hecho otros usuarios.
4. El usuario puede puntuar el ítem.
5. El usuario puede comentar el ítem.

CDU05:

Nombre:	Marcar película como vista
Descripción:	Agregar película a Lista personal de “películas vistas”
Actores:	Usuario registrado
Pre-condiciones:	Haber realizado una búsqueda de películas
Post-condiciones:	Se guarda el ítem en “películas vistas”.
Gatilladores:	El usuario selecciona la opción “visto”

Flujo Básico:

1. El usuario hace *click* sobre el botón “visto” de una película.
2. La película se agrega automáticamente a su lista personal

Flujo Alternativo:

El usuario se encuentra en la página de detalle de una película.

1. El usuario realiza una búsqueda y selecciona un título
2. El usuario entra en la página de detalle de la película
3. El usuario hace *click* sobre el botón “visto” de una película.
4. La película se agrega automáticamente a su lista personal

El usuario no ha entrado en su cuenta.

1. El usuario hace *click* sobre el botón “visto” de una película.
2. Aparece un *pop-up* avisando que debe ingresar a su cuenta.
3. El usuario ingresa a su cuenta (CDU02).
4. El usuario vuelve a la página de su búsqueda y volvemos al paso 1 del flujo básico.

CDU06:

Nombre:	Añadir película a mis películas favoritas
Descripción:	Agregar película a Lista personal de “películas Favoritas”
Actores:	Usuario registrado
Pre-condiciones:	Haber realizado una búsqueda de películas
Post-condiciones:	Se guarda el ítem en “películas Favoritas”.
Gatilladores:	El usuario selecciona la opción “Favorito”

Flujo Básico:

3. El usuario hace *click* sobre el botón “favorito” de una película.
4. La película se agrega automáticamente a su lista personal de favoritos

Flujo Alternativo:

El usuario se encuentra en la página de detalle de una película.

5. El usuario realiza una búsqueda y selecciona un título
6. El usuario entra en la página de detalle de la película
7. El usuario hace *click* sobre el botón “favorito” de una película.
8. La película se agrega automáticamente a su lista personal

El usuario no ha entrado en su cuenta.

5. El usuario hace *click* sobre el botón “favorito” de una película.
6. Aparece un *pop-up* avisando que debe ingresar a su cuenta.
7. El usuario ingresa a su cuenta (CDU02).
8. El usuario vuelve a la página de su búsqueda y volvemos al paso 1 del flujo básico.

CDU07:

Nombre:	Marcar Serie como vista
Descripción:	Agregar serie, temporada y episodios específicos a Lista personal de “series vistas”
Actores:	Usuario registrado
Pre-condiciones:	Haber realizado una búsqueda de series
Post-condiciones:	Se guarda el ítem en “series vistas”.
Gatilladores:	El usuario selecciona la opción “visto”

Flujo Básico:

1. El usuario hace *click* sobre el botón “visto” de una serie.
2. Aparecen en pantalla las temporadas y episodios de la serie.
5. El usuario selecciona los episodios vistos.
6. La serie se agrega automáticamente a su lista personal

Flujo Alternativo:

El usuario marca temporadas completas o serie completa.

1. El usuario realiza una búsqueda y selecciona un título
2. Aparecen en pantalla las temporadas y episodios de la serie.
3. El usuario hace *click* sobre el botón de una temporada o “todo”.
4. La serie se agrega automáticamente a su lista personal

El usuario no ha entrado en su cuenta.

1. El usuario hace *click* sobre el botón “visto” de una serie.
2. Aparece un *pop-up* avisando que debe ingresar a su cuenta.

3. El usuario ingresa a su cuenta (CDU02).
4. El usuario vuelve a la página de su búsqueda y volvemos al paso 1 del flujo básico.

CDU08:

Nombre:	Añadir Serie a mis series favoritas
Descripción:	Agregar serie a Lista personal de “series favoritas”
Actores:	Usuario registrado
Pre-condiciones:	Haber realizado una búsqueda de series
Post-condiciones:	Se guarda el ítem en “series favoritas”.
Gatilladores:	El usuario selecciona la opción “favorito”

Flujo Básico:

3. El usuario hace *click* sobre el botón “favorito” de una serie.
4. Aparecen en pantalla las temporadas y episodios de la serie.
7. El usuario selecciona los episodios vistos.
8. La serie se agrega automáticamente a su lista personal de favoritos

Flujo Alternativo:

El usuario marca temporadas completas o serie completa.

5. El usuario realiza una búsqueda y selecciona un título
6. Aparecen en pantalla las temporadas y episodios de la serie.
7. El usuario hace *click* sobre el botón de una temporada o “todo”.
8. La serie se agrega automáticamente a su lista personal

El usuario no ha entrado en su cuenta.

5. El usuario hace *click* sobre el botón “favorito” de una serie.
6. Aparece un *pop-up* avisando que debe ingresar a su cuenta.
7. El usuario ingresa a su cuenta (CDU02).
8. El usuario vuelve a la página de su búsqueda y volvemos al paso 1 del flujo básico.

CDU09:

Nombre:	Ver ítem en panel de usuario
Descripción:	Ver los ítems que están en las listas de visionados del usuario
Actores:	Usuario registrado

Pre-condiciones:	Haber agregado ítems a sus listas
Post-condiciones:	Se muestran las listas del usuario
Gatilladores:	El usuario selecciona la opción Mi Biblioteca

Flujo Básico:

1. El usuario selecciona su biblioteca
2. Aparecen los títulos agregados a sus listas personales
3. El usuario selecciona un ítem (CDU04)

Flujo Alternativo:

El usuario selecciona el episodio que aparece en representación de cada serie.

1. El usuario selecciona su biblioteca de series
2. El usuario elige el tipo de orden: Episodio siguiente a ver o último episodio emitido.
3. Aparecen los títulos agregados a sus listas personales
4. El usuario selecciona un ítem (CDU04)

CDU10:

Nombre:	Eliminar ítem en panel de usuario
Descripción:	Eliminar ítems que están en las listas de visionados del usuario
Actores:	Usuario registrado
Pre-condiciones:	Haber agregado ítems a sus listas
Post-condiciones:	El ítem seleccionado se elimina de la lista
Gatilladores:	El usuario selecciona la opción Borrar Ítem

Flujo Básico:

1. El usuario selecciona su biblioteca
2. Aparecen los títulos agregados a sus listas personales
3. El usuario selecciona un ítem (CDU04)
4. El usuario hace click en Borrar ítem
5. La lista se actualiza eliminando el ítem seleccionado

CDU11:

Nombre:	Agregar ítem a Watchlist
Descripción:	Agregar ítem a Lista personal de películas y series por ver
Actores:	Usuario registrado

Pre-condiciones:	Haber realizado una búsqueda de películas y/o series
Post-condiciones:	Se guarda el ítem en "Watchlist".
Gatilladores:	El usuario selecciona la opción "Watchlist"

Flujo Básico:

1. El usuario hace busca un título (CDU03).
2. El usuario hace *click* sobre el botón watchlist de algún ítem
3. El ítem se agrega automáticamente a su lista Watchlist

Flujo Alternativo:

El usuario se encuentra en la página de detalle de una película.

1. El usuario realiza una búsqueda y selecciona un título
2. El usuario entra en la página de detalle de la película
3. El usuario hace *click* sobre el botón watchlist de algún ítem
4. La película se agrega automáticamente a su lista Watchlist

El usuario no ha entrado en su cuenta.

1. El usuario hace *click* sobre el botón "watchlist" de una película.
2. Aparece un *pop-up* avisando que debe ingresar a su cuenta.
3. El usuario ingresa a su cuenta (CDU02).
4. El usuario vuelve a la página de su búsqueda y volvemos al paso 1 del flujo básico.

CDU12:

Nombre:	Etiquetar ítem en panel de usuario
Descripción:	Agregar etiquetas a los ítems que están en las listas del usuario
Actores:	Usuario registrado
Pre-condiciones:	Haber agregado ítems a sus listas
Post-condiciones:	Se agregan etiquetas a las series, temporadas, episodios y/o películas
Gatilladores:	El usuario selecciona la opción Mi Biblioteca

Flujo Básico:

1. El usuario selecciona su biblioteca
2. Aparecen los títulos agregados a sus listas personales
3. El usuario selecciona un ítem (CDU04)
4. El usuario hace *click* sobre el botón *tags*
5. El usuario escribe uno o varios *tags* separados por 'comas'.

CDU13:

Nombre:	Filtrar series y películas por tags
Descripción:	El usuario hace una consulta por un cierto tag utilizando las opciones de búsqueda habilitadas en el sitio.
Actores:	Usuario registrado, invitado
Pre-condiciones:	ninguna
Post-condiciones:	Aparecen los resultados del tag buscado
Gatilladores:	El usuario selecciona el botón buscar

Flujo Básico:

1. El usuario escribe uno o varios *tags* en buscador.
2. El usuario hace *click* en buscar.
3. El usuario puede ver información de los resultados al hacer *click* sobre ellos.

Flujo Alternativo:

El usuario está registrado y puede elegir una búsqueda en sus listas o pública.

1. El usuario escribe uno o varios *tags* en buscador.
2. El usuario hace *click* en buscar.
3. El usuario puede ver información de los resultados al hacer *click* sobre ellos.
4. El usuario selecciona si desea restringir los resultados al contenido de sus listas

El usuario hizo su búsqueda y no hay resultados.

1. Se da la opción al usuario de volver a buscar.

CDU14:

Nombre:	Dejar un review en un ítem
Descripción:	El usuario puede agregar un comentario a un ítem.
Actores:	Usuario registrado
Pre-condiciones:	Estar visualizando el detalle de un ítem
Post-condiciones:	El comentario se agrega a la página detalle del ítem

Gatilladores:	El usuario hace <i>click</i> en Enviar comentario.
---------------	--

Flujo Básico:

1. El usuario escribe un comentario en el cuadro de texto habilitado.
2. El usuario presiona enviar.

Flujo Alternativo:

El usuario cierra la ventana de visualización sin presionar el botón de enviar.

1. El mensaje se borra y no es enviado.

CDU15:

Nombre:	Puntuar un ítem
Descripción:	El usuario puede evaluar con estrellas un ítem.
Actores:	Usuario registrado
Pre-condiciones:	Estar visualizando el detalle de un ítem
Post-condiciones:	La puntuación se agrega a las estadísticas del ítem
Gatilladores:	El usuario hace <i>click</i> en la cantidad de estrellas.

Flujo Básico:

1. El usuario selecciona una cantidad de estrellas en la zona habilitada.

Flujo Alternativo:

El usuario edita su puntuación.

1. El usuario selecciona una cantidad de estrellas diferente a la ya escogida
2. La puntuación del ítem cambia

CDU16:

Nombre:	Puntuar un review
Descripción:	El usuario puede evaluar con estrellas el review de otro usuario.
Actores:	Usuario registrado
Pre-condiciones:	Estar visualizando el detalle de un ítem

Post-condiciones:	La puntuación se agrega a las estadísticas del review
Gatilladores:	El usuario hace <i>click</i> en la cantidad de estrellas.

Flujo Básico:

1. El usuario selecciona una cantidad de estrellas en la zona habilitada.

Flujo Alternativo:

El usuario edita su puntuación.

1. El usuario selecciona una cantidad de estrellas diferente a la ya escogida
2. La puntuación del review cambia

CDU17:

Nombre:	Crear una publicación
Descripción:	El usuario puede compartir contenido con otros usuarios a través de las publicaciones
Actores:	Usuario registrado
Pre-condiciones:	Ser un usuario registrado
Post-condiciones:	Se crea una nueva publicación en el sistema
Gatilladores:	El usuario hace <i>click</i> en crear publicación

Flujo Básico:

2. El usuario selecciona *crear publicación*
3. El usuario escribe y adjunta los contenidos que desee en la publicación
4. El usuario especifica la película o serie (en este caso con temporada y/o episodio)
5. Agrega los *tags* que estime necesarios
6. El usuario hace *click* en *enviar*

Flujo Alternativo:

El usuario cierra la ventana de visualización sin presionar el botón de enviar.

1. El mensaje se borra y no es enviado.

CDU18:

Nombre:	Acceder al <i>dashboard</i>
Descripción:	El usuario tras entrar a su cuenta, accede a su <i>dashboard</i> personal donde aparecen las publicaciones de sus amigos y de otros usuarios. Las publicaciones públicas se filtran por las series que ya están en sus listas, mientras que las de sus amigos pueden incluir otros ítems.

Actores:	Usuario registrado
Pre-condiciones:	Ser un usuario registrado
Post-condiciones:	Se muestra el dashboard del usuario
Gatilladores:	El usuario accede a su cuenta

Flujo Básico:

1. El usuario accede a su cuenta (CDU02)
2. El usuario entra a su *dashboard*
3. El usuario especifica si desea filtrar las publicaciones para solo ver las de sus amigos
4. El usuario navega por las publicaciones

Flujo Básico:

El usuario filtra por *tags* su *dashboard*.

1. El usuario accede a su cuenta (CDU02)
2. El usuario entra a su *dashboard*
3. El usuario especifica si desea filtrar las publicaciones para solo ver las de sus amigos
4. El usuario ingresa *tags* en el filtro
5. El usuario navega por las publicaciones

CDU19:

Nombre:	Excluir contenido del <i>dashboard</i>
Descripción:	El usuario puede seleccionar <i>tags</i> , películas, series, temporadas o episodios específicos para que todas las publicaciones que los tengan en sus temáticas avisen la situación y así evite <i>spoilers</i> .
Actores:	Usuario registrado
Pre-condiciones:	Ser un usuario registrado
Post-condiciones:	Se muestra el dashboard del usuario sin <i>spoilers</i>
Gatilladores:	El usuario accede a su cuenta

Flujo Básico:

1. El usuario accede a su cuenta (CDU02)
2. El usuario entra a su perfil
3. El usuario hace *click* en *spoilers*
4. El usuario especifica todos los ítems que considera spoiler
5. El usuario hace *click* en *guardar configuración*

CDU20:

Nombre:	Seguir a un usuario
Descripción:	El usuario puede seguir a otro usuario (ser su amigo) y recibir las publicaciones de este en su <i>dashboard</i>
Actores:	Usuario registrado
Pre-condiciones:	Ser un usuario registrado
Post-condiciones:	Se muestra el dashboard del usuario sin <i>spoilers</i>
Gatilladores:	El usuario accede a su cuenta

Flujo Básico:

1. El usuario accede a su cuenta (CDU02)
2. El usuario entra a su perfil
3. El usuario busca por nombre a otro usuario
4. Entra a su página de perfil y selecciona *seguir*

Flujo Alternativo:

El usuario encuentra amigos a través de *reviews* o publicaciones públicas.

1. El usuario hace *click* sobre el nombre del usuario al que quiere seguir
2. Entra a su página de perfil y selecciona *seguir*

CDU21:

Nombre:	Compartir una publicación o ítem
Descripción:	El usuario puede compartir la información de un ítem o publicación en otras redes sociales
Actores:	Usuario registrado, invitado
Pre-condiciones:	ninguna
Post-condiciones:	Se comparte la información en otra red social
Gatilladores:	El usuario hace <i>click</i> en <i>compartir</i>

Flujo Básico:

1. El usuario selecciona el botón *compartir*
2. El usuario elige la red social de interés

Diseño Web

Aspectos generales

Lenguajes de programación web utilizados:

- Ruby 2.0
- Rails 4.0
- CSS/SCSS
- HTML
- JavaScript con jQuery

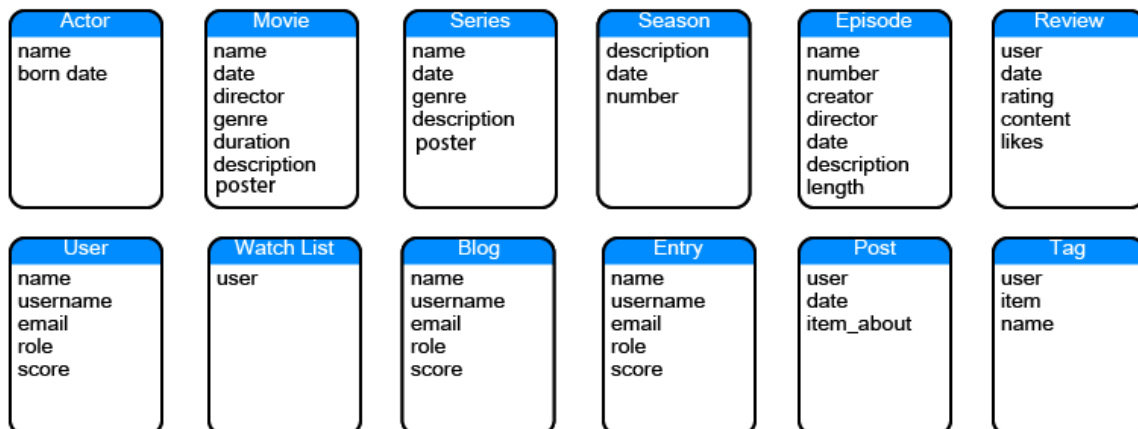
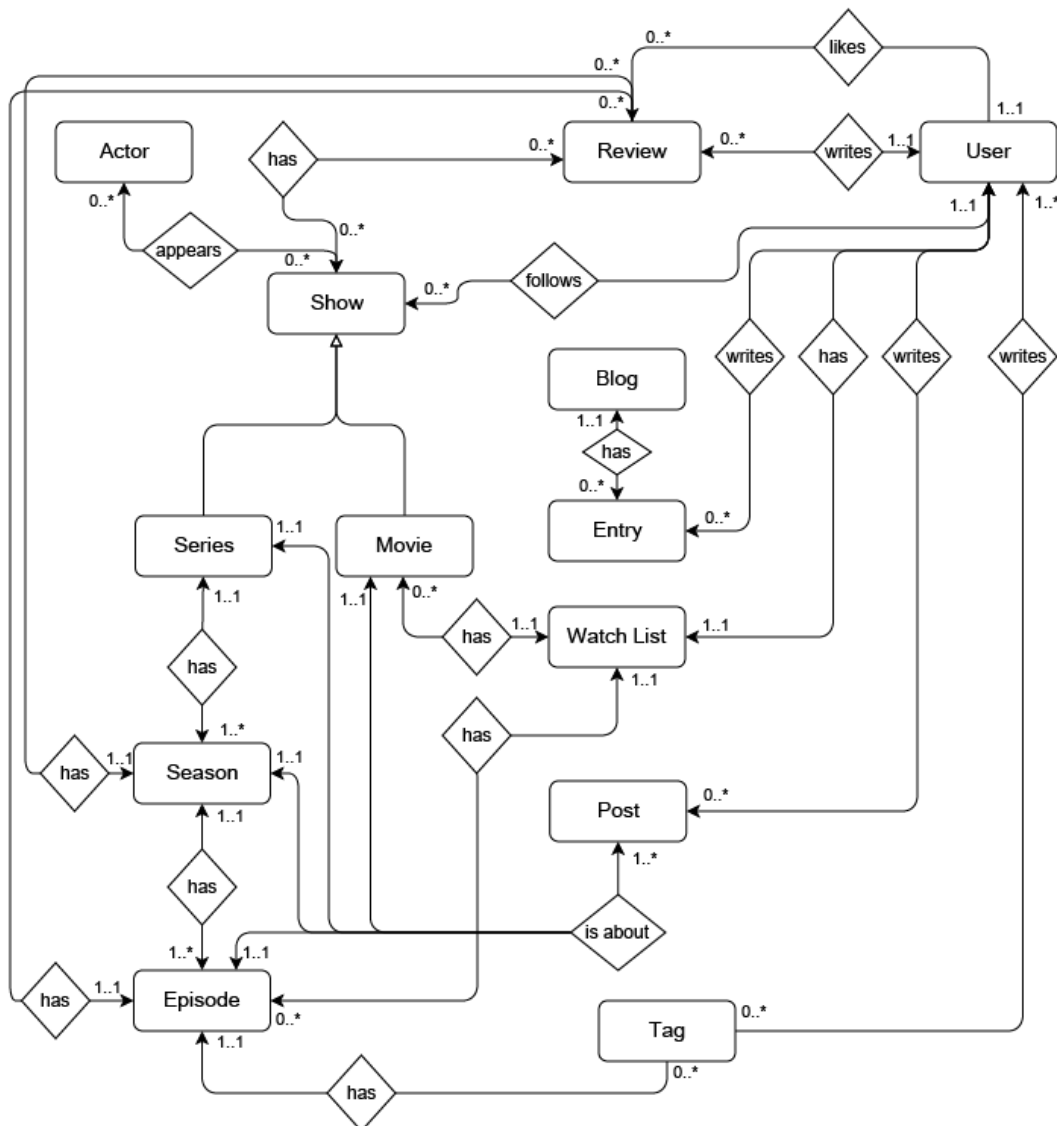
Aplicaciones utilizadas:

- **Heroku**: testeo de aplicación en producción
- **Gmail**: envío de mails, cuenta showganizer@gmail.com con la password **showganizer-www**.
- **Dropbox**: almacenamiento de imágenes en posts, se utiliza la cuenta de gmail asociada, con la misma contraseña de acceso.
- **Rottentomatoes**: uso de api para rating de películas y poster de películas
- **Trakt**: uso de api para poster de series y autocompletado de form al crear un nuevo show.

Estructura de la aplicación:

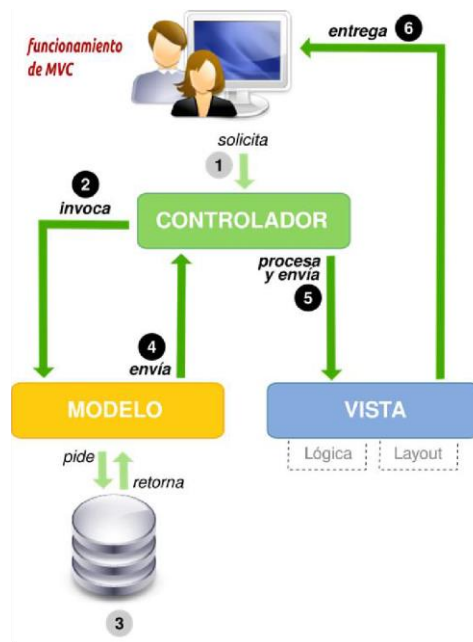
- Zona de usuarios: el usuario puede revisar su watchlist, sus favoritos o la lista con los shows que ha visto y/o está viendo.
- Catálogo de shows: el usuario o visitante puede buscar shows e información de los mismos
- Dashboard: el usuario puede ver posts sobre sus shows favoritos.

Modelo de datos



Modelo de Arquitectura

La plataforma web posee una arquitectura basada en MVC (Modelo Vista Controlador) debido a que está construida con Ruby on Rails, y es la arquitectura que ofrece.



Modelos

Referidos a los datos y generalmente representados por una tabla en la base de datos. En Rails, se crea una tabla para cada modelo, y éste es el responsable de definir las relaciones con otros modelos. Así por ejemplo, el Season, tendrá una Serie. Ambos son modelos y estarán representados por tablas en la base de datos. Aunque hay modelos que no tienen su correspondiente en la base de datos.

Modelo	Tabla en BD	Datos o relaciones	Detalles
User	Users	-	-
Show	Shows	-	Single Table Inheritance
Movie	-	-	hereda de Show
Serial	-	-	hereda de Show
Show_user	ShowUsers	relaciona users con shows (mis shows)	-
Season	Seasons	--	-
Episode	Episodes	-	-
Watchlist	Watchlists	guarda shows que quiero ver	-

Show_watchlist	ShowWatchlists	relaciona shows con watchlist, cada user tiene una watchlist	-
Watched	Watcheds	Guarda movies y episodios vistos	-
Movie_user	MoviesUsers	relaciona watched con movies	-
Episode_user	EpisodeUsers	relaciona watched con episodios	-
Review	Reviews	-	pertenecen a un usuario
Post	Posts	-	Pertenecen a un usuario
Tag	Tags	-	Polymorphic association
Post_tagging	PostTaggings	relació entre tags y post	-
Episode_tagging	EpisodeTaggings	relación entre tags y episode	-
User_episode_tagging	UserEpisodeTaggings	relación entre user, episode y tags	para generar listas de usuario con el contenido

Situaciones especiales:

STI

Dado la similitud entre las entidades **Movie** y **Serial**, se decidió usar [Single Table Inheritance](#), conservando todas las filas de las entidades en una sola tabla, que representa al modelo **Show**, clase de las cuales heredan las mencionadas en un principio.

Polymorphic associations

Ya que los **review** y **posts** están relacionados con **movie**, **serial**, **season** y **posts**, se definen asociaciones polimórficas **reviewable** y **commentable**, manteniendo todas las entradas de los **reviews** en una tabla, así como también los **posts**.

Concerns

Se utilizan concerns en tres diferentes aspectos de la aplicación, los cuales también tienen distintas funciones. En las rutas se usan concerns para agregar los **reviews** y **posts** a los otros recursos de forma anidada.

En los controladores se agrupan las acciones **show_reviews** y **show_posts** para mostrar los recursos a través de todas las vistas.

En los modelos se agrupan las relaciones **has_many reviews** y **posts** para los diferentes controladores.

Seeds

El proyecto cuenta con un archivo **seeds.rb** que creará cerca de 30 películas y 2 series con varias temporadas y episodios incluidos. De manera de partir con una base de datos amplia para probar las diferentes funcionalidades.

Encapsulación de funcionalidad en modelos

Modelo	función	objetivo
Movie	search_rating(title)	generar un request de la api de Rottentomatoes para encontrar el rating de una película
Review	stars	asignar la cantidad de estrellas a un review
Serial	search_poster(title)	generar un request de la api de trakt.tv para encontrar la url de la imagen de una serie
User	gravatar_url(size)	recupera el avatar del usuario en gravatar si es que existe
User	get_access_token	genera un token para el user
User	score_update(parent)	actualiza el puntaje del user
Watched	next_episode(watched,season)	analiza cuál es el siguiente episodio de una serie que user debe ver
Episode	user_tags_list	Maneja la lista de tags de un user

Controllers

Una vez que les llega una petición desde un navegador, reflejada en la URL, se ponen en marcha realizando operaciones de manipulación de los datos descritos en los modelos: Creación, Actualización, Eliminación o Búsqueda, operaciones típicas de cualquier sistema interactivo (Denominado CRUD o Create, Retrieve, Update, Delete). Una vez realizada la operación, el controlador devuelve una respuesta que generalmente se refleja en una "Vista" o página.

Controlador	Actions añadidas (diferentes a CRUD)
Application_controller	<ul style="list-style-type: none">• set_parent: localiza los parents para las entidades polimórficas• current_user: setea al usuario actual

	<ul style="list-style-type: none"> ● authorize: filtro para ver si es usuario ● user_admin: filtro para ver si es admin
Users_controller	-
Sessions_controller	-
Movies_controller	<ul style="list-style-type: none"> ● add: agrega la movie a la lista de favoritos del usuario ● remove: saca la movie de la lista de favoritos del usuario ● watch: agrega la movie a la watchlist del usuario ● unwatch: saca la movie de la watchlist del usuario ● seen: agrega la movie a la lista watched (vistos) ● unseen: saca la movie de la lista watched (Vistos)
Serials_controller	<ul style="list-style-type: none"> ● add: agrega la serie a la lista de favoritos del usuario. ● remove: saca la serie de la lista de favoritos del usuario. ● watch: agrega la serie a la watchlist del usuario. ● unwatch: saca la serie de la lista del usuario.
Seasons_controller	-
Episodes_controller	<ul style="list-style-type: none"> ● seen: agrega el episodio e a la lista watched (vistos) ● unseen: saca el episodio e a la lista watched (vistos) ● add_user_tag: agrega un tag personal en un episodio ● remove_user_tags: quita un tag personal en un episodio
Washlists_controller	<ul style="list-style-type: none"> ● my_watchlist: renderea la vista show respectiva, que muestra el contenido de la watchlist de un user
Watched_controller	<ul style="list-style-type: none"> ● my_watched: renderea la vista show respectiva, que muestra el contenido de la lista watched de un user, separado por lo visto (series y películas completas) y lo que está viendo (series con episodios sin ver) ● tracking: renderea la vista tracking, para lo que necesita una serie en particular. Muestra el detalle de las temporadas indicando episodios siguiente. ● detail: renderea la vista detail respectiva, necesita una season de cierta serie ya que muestra su detalle. ● set_seasons: procesa las temporadas que el user ha visto completo en una lista. ● set_series: procesa en una lista las series que el user ha visto por completo. ● set_watching: crea una lista con las series que está viendo el user.
Reviews_controller	-
Posts_controller	-
Pages_controller	-

Detalles importantes

Vistas

Es lo que ve el usuario: el interfaz del sistema a través del cual realiza peticiones.

Conjunto Vista	Vistas que contiene
Users	<ul style="list-style-type: none">• index: lista de los usuarios registrados• show: perfil del usuario• _form: formulario de nuevo usuario• _form_edit: formulario para editar el usuario• new: vista que muestra el formulario de nuevo usuario• edit: lista que muestra el formulario de edición de usuario-
Sessions	<ul style="list-style-type: none">• new: formulario para iniciar sesión en la aplicación
Movies	<ul style="list-style-type: none">• _action_menu: reúne los botones de acción• _admin_bar: reúne los links para la barra de administración• _content: contenido de la movie (título y descripción)• _form: formulario para crear y editar películas.• _sidebar: contenido de la barra lateral• edit: muestra el formulario para edición.• index: vista que muestra una lista de las películas• new: muestra el formulario para crear una nueva película• show_posts: versión de la vista show con los posts desplegados• show_reviews: versión de la vista show con los reviews desplegados• show: vista que muestra una película en específico (muestra los reviews por defecto)
Serials	<ul style="list-style-type: none">• _action_menu: reúne los botones de acción• _admin_bar: reúne los links para la barra de administración• _content: contenido de la serial (título y descripción)• _form: formulario para crear y editar serials.• _sidebar: contenido de la barra lateral• edit: muestra el formulario para edición.• index: vista que muestra una lista de las serials.• new: muestra el formulario para crear una nueva serial• show_posts: versión de la vista show con los posts desplegados• show_reviews: versión de la vista show con los reviews desplegados• show: vista que muestra una serial en específico (muestra los reviews por defecto)
Seasons	<ul style="list-style-type: none">• _action_menu: reúne los botones de acción• _admin_bar: reúne los links para la barra de administración• _content: contenido de la season (título y descripción)• _form: formulario para crear y editar season.• _sidebar: contenido de la barra lateral• edit: muestra el formulario para edición.• index: vista que muestra una lista de las seasons.• new: muestra el formulario para crear una nueva season.• show_posts: versión de la vista show con los posts desplegados• show_reviews: versión de la vista show con los reviews desplegados• show: vista que muestra una season en específico (muestra los reviews

	por defecto)
Episodes	<ul style="list-style-type: none"> • <code>_action_menu</code>: reúne los botones de acción • <code>_admin_bar</code>: reúne los links para la barra de administración • <code>_content</code>: contenido del episode (título y descripción) • <code>_form</code>: formulario para crear y editar serials. • <code>_sidebar</code>: contenido de la barra lateral • <code>_tag_form</code>: formulario para agregar etiquetas al episode • <code>_user_tag_list</code>: vista que muestra las etiquetas propias del usuario para el episodio • <code>edit</code>: muestra el formulario para edición. • <code>index</code>: vista que muestra una lista de las serials. • <code>new</code>: muestra el formulario para crear una nueva serial • <code>show_posts</code>: versión de la vista show con los posts desplegados • <code>show_reviews</code>: versión de la vista show con los reviews desplegados • <code>show</code>: vista que muestra una serial en específico (muestra los reviews por defecto)
Watchlists	<ul style="list-style-type: none"> • <code>show</code>: muestra todos los shows de la watchlist del user
Watched	<ul style="list-style-type: none"> • <code>my_watched</code>: shows de la lista watched del user • <code>tracking</code>: detalle de tracking de una serie • <code>detail</code>: detalle detracking de una season
Reviews	maneja las vistas parciales de reviews
Posts	maneja las vistas parciales de reviews
Layouts	<ul style="list-style-type: none"> • <code>_admin_bar</code> panel: panel de administrador • <code>_flash_bar</code>: barra de aviso • <code>_main_nav</code>: navegación principal • <code>_top_search</code>: buscador • <code>_top_user_bar</code>: panel de usuario
Pages	<ul style="list-style-type: none"> • <code>index</code>: home de la aplicación • <code>no_access</code>: página 403
User_mailer	<ul style="list-style-type: none"> • <code>welcome_mail</code>: email de bienvenida • <code>new_episode</code>: aviso de nuevo episodio

Detalles

Partials

Cada directorio de las vistas considera diferentes **partials** para los diferentes secciones de las vistas principales, así los archivos se separan de forma modular. Al mismo tiempo se crean partials con el nombre del recurso, que se utilizan para iterar al hacer **render** de una colección, de esta forma la vista es consistente a lo largo de la aplicación, esto último se utiliza fuertemente en las entidades de **posts** y **reviews**.

URLS no intuitivas

Ruta	vista
Home	Pages/index
403	Pages/no_access
/me	Users/show
/me/watchlist	Watchlist/show
/me/watchlist/serials/:id	Watched/tracking
/me/watchlist/serials/:serial_id/seasons/:season_id	Watched/detail
/me/seen	Watched/my_watched

Gemas relevantes

paperclip 3.0: para manejar la subida de archivos, en este caso imágenes (gif, jpg y png). Su configuración se encuentra en el **modelo Post**, ya que cada Post puede tener una imagen.

paperclip-dropbox 1.1.7: para almacenar las imágenes que se suben a través de paperclip en una cuenta de dropbox. El archivo **dropbox.yml** de **configuración** contiene la información necesaria (datos personales) de la cuenta usada.

Organización de los estilos

Archivo CSS	vistas que afecta
Application	Todas. Características generales
Indexgrid	Diseño para mostrar películas agrupadas. <ul style="list-style-type: none">● followed_shows\movies● followed_shows\serials● followed_shows\shows● movies\index● watched\show● watchlists\show
Movies	Diseño del rating de una película <ul style="list-style-type: none">● movies_sidebar
Posts	Vistas de Post

Reviews	Vistas de Review
Session	Sessions\new
Showgrid	<ul style="list-style-type: none"> • movies\show • serials\show • episodes\show
User	Vistas de User

APIS

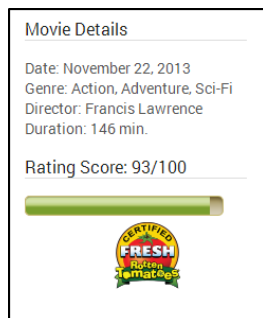
Consumidas

Para generar los requests se utilizó el módulo **net/http** de Ruby.

Rottentomatoes, <http://developer.rottentomatoes.com/io-docs>

Se utilizó el método **Movies search** de la api, para realizar búsquedas de películas y recibir el rating de la audiencia y su respectivo poster. Para ello se hacía un request como el siguiente:

```
http://api.rottentomatoes.com/api/public/v1.0/movies.json?apikey=apikey?q=titl
e
```



Agregando la key correspondiente y el query del título de una película, para ello se usa el atributo name del modelo show. Se recibe un json y se extraen los datos mencionados anteriormente. El código de la api se encuentra como método en el modelo Movie.

Trakt, <http://trakt.tv/api-docs>

En este caso, se utilizó el método **Search/shows** y **Search/movies**. El primero, busca series con el título, se utiliza en dos secciones de la aplicación, primero en el modelo de **serials** para poder obtener una url del poster de la serie y también, en **main.js**, donde a través de javascript y ajax rellenamos algunos campos del formulario para crear una nueva serie automáticamente; la acción anterior se repite en el formulario para crear una nueva película, a través de search/movies.

```
http://api.trakt.tv/search/shows.json?apikey=query=title
```


El proceso es el mismo que en la api de Rottentomatoes, se envía un request con el título de la película o serie y se reciben los datos para luego extraer lo que nos interesa como es descripción, nombre oficial y url de poster.

New movie

[Click here to fill the fields](#)

```
{
  "title": "The Big Bang Theory",
  "year": 2007,
  "url": "http://trakt.tv/show/the-big-bang-theory",
  "first_aired": 1188630000,
  "country": "United States",
  "overview": "Leonard Hofstadter and Sheldon Cooper are brilliant physicists,",
  "runtime": 20,
  "network": "CBS",
  "air_day": "Thursday",
  "air_time": "8:00pm",
  "certification": "TV-PG",
  "imdb_id": "tt0898266",
  "tvdb_id": 80379,
  "tvrage_id": 8511,
  "ended": false, //has the show ended?
  "images": {
    "poster": "http://trakt.us/images/posters/34.jpg",
    "fanart": "http://trakt.us/images/fanart/34.jpg",
    "banner": "http://trakt.us/images/banners/34.jpg"
  },
  "genres": ["Action", "Comedy"],
  "seasons": [ //optional
    {
      "season": 6,
      "episodes": [
        {
          "season": 6,
          "number": 24
        }
      ]
    }
  ]
}
```

Propias

Se construyó una API propia que cumple con las siguientes funciones:

Películas

- Listar todas las películas, listar una película en específico, listar sus reviews y posts.
- Agregar/remover de mis películas, agregar/remover de mi watchlist y marcar como vista/no vista. (requiere Access Token)
- Crear y eliminar una película (solo administradores)

Series

- Listar todas las series, listar una serie en específico, listar todas las temporadas de una serie, listar una temporada en específico, listar todos los episodios de una temporada y listar un episodio en específico.

- Para las series, temporadas y episodios: listar los reviews y posts; y crear eliminar cada uno de estos (solo administradores).
- Para las series: Agregar/remover de mis series, agregar/remover de mi watchlist.
- Para los episodios: marcar como visto/ no visto.

RESTful API

Para las funciones que requiere un usuario asociado, se debe incluir el siguiente header en la consulta: Authorization: Token token="token_string", donde token_string corresponde al access token, obtenible por el usuario en su perfil.

Los siguientes ejemplos usan cURL para gerenciar las consultas, en las cuales se expresa el método HTTP utilizado, se mostrarán solo los casos para las películas, luego se adjunta la lista completa de consultas.

Listar Películas

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies
```

Resultado:

```
[
  {
    id: 6,
    name: "Serenity",
    date: "2013-10-10T01:39:00.000Z",
    genre: " Action, Adventure, Sci-Fi",
    description: "The crew of the ship Serenity tries to evade an assassin sent to recapture one of their number who is telepathic. Remember, you can't take the sky from me.",
    director: "Joss Whedon",
    duration: 119,
    created_at: "2013-10-10T01:40:49.415Z",
    updated_at: "2013-12-01T17:48:57.143Z",
    poster: "http://content6.flixster.com/movie/11/17/07/11170768_ori.jpg"
  },
  {
    id: 2,
    name: "Jackass 2.5",
    date: "2013-10-03T21:39:00.000Z",
    genre: " Documentary, Comedy ",
    description: "The crew have now set off to finish what as left over from Jackass 2.0, and in this version they have Wee Man use a 'pee' gun on themselves, having a mini motor bike fracas in the grocery mall, a sperm test, a portly crew member disguised as King Kong, as well as include three episodes of their hilarious adventures in India, namely drinking beer off of Shridhar Chillai's several feet long fingernails; having one of the crew lie on a bed of nails with two snakes - one on his chest and one between the legs, as well as a decorated elephant in the background; and finally having a half-naked Indian Sadhu drink one of the crew's urine. ",
    director: "Jeff Tremaine",
    duration: 64,
    created_at: "2013-10-03T21:40:01.894Z",
    updated_at: "2013-12-01T17:48:53.082Z",
    poster: "http://content7.flixster.com/movie/10/88/83/10888301_ori.jpg"
  },
  {
    id: 7,
    name: "The Hunger Games: Catching Fire",
    date: "2013-11-22T01:45:00.000Z",
    genre: "Action, Adventure, Sci-Fi",
    description: "Katniss Everdeen and Peeta Mellark become targets of the Capitol after their
```

```
victory in the 74th Hunger Games sparks a rebellion in the Districts of Panem.",
  director: " Francis Lawrence",
  duration: 146,
  created_at: "2013-10-10T01:46:21.523Z",
  updated_at: "2013-12-01T17:49:01.233Z",
  poster: "http://content7.flixster.com/movie/11/17/38/11173861_ori.jpg"
}
```

Listar película por su id

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies/id
```

Resultado

```
{
  id: 6,
  name: "Serenity",
  date: "2013-10-10T01:39:00.000Z",
  genre: " Action, Adventure, Sci-Fi",
  description: "The crew of the ship Serenity tries to evade an assassin sent to recapture one of
their number who is telepathic. Remember, you can't take the sky from me.",
  director: "Joss Whedon",
  duration: 119,
  created_at: "2013-10-10T01:40:49.415Z",
  updated_at: "2013-12-01T17:48:57.143Z",
  poster: "http://content6.flixster.com/movie/11/17/07/11170768_ori.jpg"
}
```

Listar reviews de película

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/reviews
```

Listar posts de película

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/posts
```

Agregar/remover película de mis películas

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/add -H
'Authorization: Token token="token_string"'

curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/remove -H
'Authorization: Token token="token_string"'
```

Marcar como vista/no vista

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/seen -H
'Authorization: Token token="token_string"'

curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/unseen -H
'Authorization: Token token="token_string"'
```

Agregar/remover de watchlist

```
curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/watch -H
'Authorization: Token token="token_string"'

curl -X GET http://showganizer-www.herokuapp.com/api/movies/id/unwatch -H
'Authorization: Token token="token_string"'
```

Crear/eliminar/actualizar una película

```
curl -X POST http://showganizer-www.herokuapp.com/api/movies/ -d DATA -H
'Authorization: Token token="token_string"'

curl -X DELETE http://showganizer-www.herokuapp.com/api/movies/id/ -H
'Authorization: Token token="token_string"'

curl -X PATCH http://showganizer-www.herokuapp.com/api/movies/id/ -d DATA -H
'Authorization: Token token="token_string"'
```

Lista completa

Método	URL	Requisitos
GET	/api/movies	
POST	/api/movies	DATA Authorization
GET	/api/movies/ id	
GET	/api/movies/ id /add	Authorization
GET	/api/movies/ id /remove	Authorization
GET	/api/movies/ id /watch	Authorization
GET	/api/movies/ id /unwatch	Authorization
GET	/api/movies/ id /seen	Authorization

GET	/api/movies/ id /unseen	Authorization
GET	/api/movies/ id /reviews	
GET	/api/movies/ id /posts	
GET	/api/series	
POST	/api/series	DATA Authorization
GET	/api/series/ id	
GET	/api/series/ id /add	Authorization
GET	/api/series/ id /remove	Authorization
GET	/api/series/ id /watch	Authorization
GET	/api/series/ id /unwatch	Authorization
GET	/api/series/ id /reviews	
GET	/api/series/ id /posts	
GET	/api/series/ id /seasons	
POST	/api/series/ id /seasons	DATA Authorization
GET	/api/series/ id /seasons/ number	
GET	/api/series/ id /seasons/ number /reviews	
GET	/api/series/ id /seasons/ number /posts	
GET	/api/series/ id /seasons/ number /episodes	
GET	/api/series/ id /seasons/ number /episodes/ number	

GET	/api/series/ id /seasons/ number /episodes/ number /reviews	
GET	/api/series/ id /seasons/ number /episodes/ number /posts	
GET	/api/series/ id /seasons/ number /episodes/ number /seen	Authorization
GET	/api/series/ id /seasons/ number /episodes/ number /unseen	Authorization

Mailers

En la plataforma web se utiliza un *mailer* para enviar dos tipos de mensajes por medio del mail: Bienvenida tras registro y aviso de nuevo episodio de una serie favorita. Para el primer evento, se indica la orden de **deliver** en la acción **create** de **user_controller**; mientras que para el aviso de nuevo episodio, este se realiza en la acción **create** de **episode_controller**.

- UserMailer.welcome_email(@user).deliver
- UserMailer.new_episode(@user).deliver

La configuración de los datos de la cuenta con la que se envían los mails se realiza en **environment.rb** de **configuración**. En este caso, utilizamos una cuenta gmail.

El contenido de cada tipo de mail se modifica en la **vista user_mailer**, la cual contiene dos versiones para cada tipo de mail, una en html y la otra en texto plano.

Manejo de Sesión

El manejo de sesión se realiza a través del controlador **session**, usando como identificadores el **username** o el **e-mail** del usuario.

Javascript

El uso de javascript se enfoca principalmente en la funcionalidad AJAX, a través del parámetro data-remote, con el cual se envían respuestas JavaScript disponible en las vistas. Estas respuestas son mayoritariamente renderizado de elementos en el DOM de forma dinámica. Se trabajó principalmente en el archivo **main.js**.

Helpers

Los Helpers que se han escrito se enfocan principalmente a abstraer repetición en las vistas, especialmente navegación y links que dependen de cierto estado de la aplicación, por lo tanto se encapsulan en helpers que manejan la condicionalidad de los trozos de vista.

Requisitos para correr la aplicación

(datos necesarios en algún archivo de configuración, por ejemplo)

Tener instalado **Ruby 2.0** y **rails 4.0** es el único requisito, sumado a la instalación de las gemas mencionadas en este documento, las cuales se actualizan con el comando *bundle install*. Por otro lado, los seeds que se incluyen generan películas y series, pero no contienen usuarios, por lo tanto, para poder generar el primer usuario, es necesario registrarse y luego por consola volver al primer usuario administrador. Basta con correr en consola los siguientes comandos tras el registro de una cuenta.

```
u = User.first
u.role = "admin"
u.save
```