

Visor Interactivo

Primera práctica de G.D.I.E.

A. Antonio Boutaour Sanchez y Bernat Pericàs Serra

43202227Q i 43212796M

# Descripción de todas las funcionalidades de acceso e interacción desarrolladas en la práctica, y tecnologías utilizadas para tal fin.

Este visor interactivo permite disfrutar de manera dinámica de la exitosa serie de televisión Padre de Familia (o Family Guy en inglés). A continuación, se describirá cada funcionalidad las cuales, si no se especifica lo contrario, han sido implementadas en *JavaScript* sobre *HTML*.



Ilustración 1. Visor

La primera funcionalidad disponible es el propio visor (Ilustración 1. Visor), el cual cuenta con los clásicos botones para cambiar volumen, avanzar y retroceder por el video, detenerlo, pausarlo, activar los subtítulos, etc. Personalizados con la estética de la propia serie. Obsérvese la cabeza de Peter, el personaje principal, como puntero de la barra de control de progreso. Para las acciones de los botones se ha usado *JQuery*, librería de *JavaScript*.

A continuación, el espectador cuenta con una matriz de personajes (Ilustración 2. Matriz de personajes) los cuales aparecerán difuminados si no aparecen en la escena que este reproduciéndose en ese momento. Además, existe una función en esta matriz, si se selecciona alguno de los personajes, el continuará reproduciéndose en una escena aleatoria en la que aparezca el personaje seleccionado.

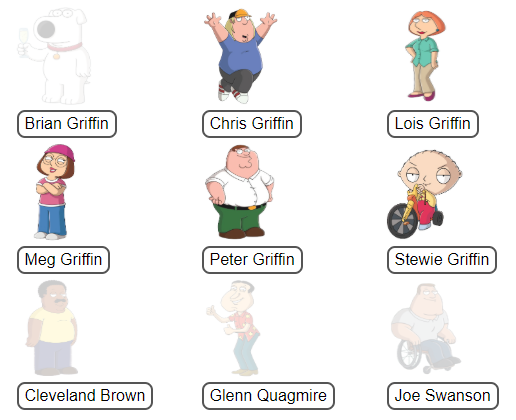


Ilustración 2. Matriz de personajes

La siguiente función (Ilustración 3. Búsqueda por Tag) también conlleva un salto en el vídeo, en concreto, al momento del video que se ha etiquetado con el tag (o etiqueta) buscado. De esta manera se puede saltar al momento del video relacionado con una idea, una frase o una palabra, aunque en el diálogo no aparezca esa palabra.

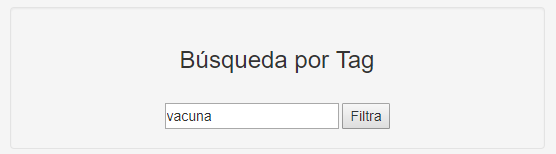


Ilustración 3. Búsqueda por Tag

En el espacio inferior de la búsqueda por Tag hay dos reproductores de audio que reproducen la canción original de la serie, y la versión acapella interpretada por los mismos autores. Véase la Ilustración 4. Canción original y a capela.

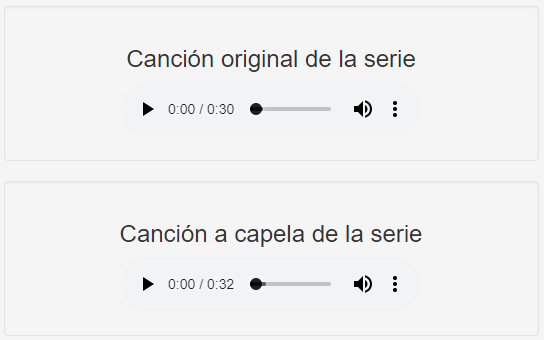


Ilustración 4. Canción original y a capela

Padre de familia se caracteriza por la abundancia de gags durante el transcurso de cada episodio, estos son pequeños videos graciosos que complementan la historia que se cuenta en un momento dado. Son tan populares que llegan a tener más protagonismo que el propio hilo argumental del episodio, así que hemos dotado al visor de una manera para valorar el episodio a partir de las puntuaciones que el usuario da a cada gag en un menú que aparece al final de cada uno de estos. Como se aprecia en la Ilustración 5. Modal para puntuar gags.



Ilustración 5. Modal para puntuar gags

El usuario valora de 1 a 5 estrellas ese gag y se guardan las puntuaciones en un lugar específico de la página como se puede ver en la siguiente ilustración (Ilustración 6. Puntuaciones de los gags).

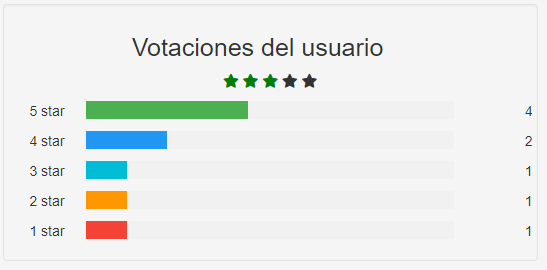


Ilustración 6. Puntuaciones de los gags

Como toda página para visualizar contenido audiovisual, cuenta con un modo oscuro, o modo noche, para no perjudicar la vista. Se activa y desactiva clicando un botón en la parte superior de la página, el cual simplemente cambia los colores de fondo de los principales elementos. Se puede ver el efecto en la Ilustración 7. Modo oscuro.

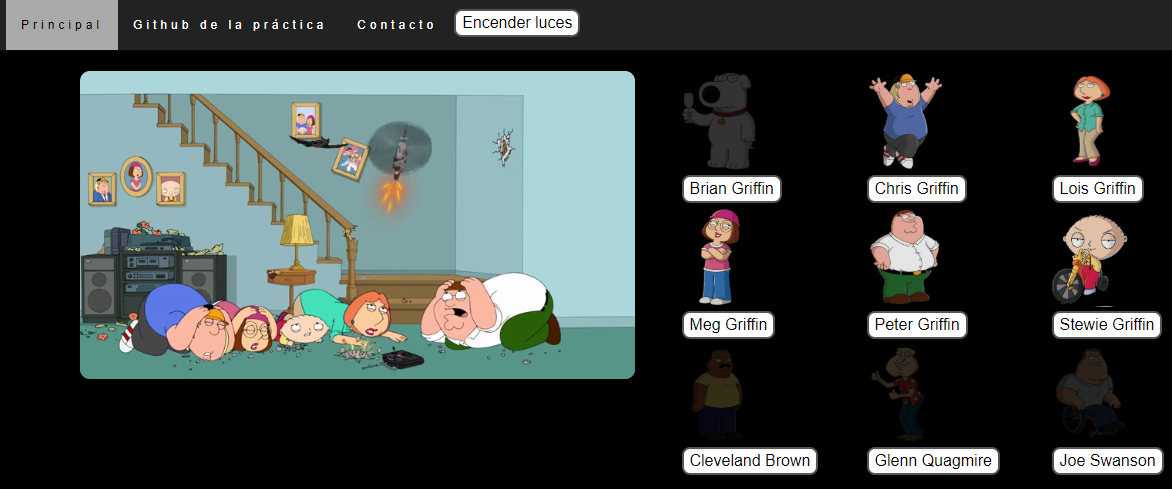


Ilustración 7. Modo oscuro

Finalmente, hay pequeñas funcionalidades extras como, por ejemplo, pausar o reproducir el vídeo haciendo clic sobre este y no exclusivamente sobre el botón de Play/Pause. O también que al hacer clic sobre la pestaña “Contacto” (parte superior de la página), se redirige al usuario a la misma página, pero al elemento con los datos de contacto de los autores de la página y se ilumina éste por un tiempo de dos segundos. Funcionalidades que, si bien no afectan al uso general de la web, si mejoran directamente la experiencia del usuario.

# Explicación técnica del proceso de codificación y publicación del material multimedia.

Se ha usado el software WebStorm para codificar la práctica, para administrar el repositorio de Git y para publicarla al servidor de la asignatura (mediante una conexión *FSTP*).

Al haberse hecho grupo, optamos por usar Git para poder trabajar a la vez, dividimos la página en diversas partes y funcionalidades y cada componente desarrollaba lo que le correspondía en su rama, una vez finalizado el desarrollo de una funcionalidad, ambos usuarios la testaban y mejoraban (si era el caso) y se hacía *pull* a la rama máster.

El principio del desarrollo de la web, el testeo de ésta se hizo en local, pero una vez un componente del grupo configuró el servidor y el mismo WebStorm para realizar la carga a éste, se pasó a testearse siempre en línea. De esta manera nos ahorramos algunos errores de Chrome al no cargar ficheros .vtt de una web en local. A demás, si el resultado se quiere en línea, no hay razones para no realizar todas las pruebas a la web ya en línea.

# Explicación técnica del uso de TextTracks, códigos JavaScript, JSONs, etc. utilizados para desarrollar todas las funcionalidades (metadata, capítulos, ...) y aspectos de desarrollo que sean relevantes.

El principal funcionamiento de la página web respecto a metadata se basa en la búsqueda de escenas, identificación de los personajes que interactúan en la escena y el servicio de rating que ofrece.

Contamos con un VTT que contiene la información separada por escena, con tiempo de inicio y fin, tags, y los personajes que intervienen en la escena. Dentro de nuestro funcionamiento del *JavaScript* encontramos un evento que se ejecuta cada vez que se realiza un cambio de Cue. Dentro de este evento se realiza un apagado de todas las imágenes de los personajes y se activan aquellas imágenes que están definidas dentro del cue. Este mismo archivo es usado para ser recorrido cuando se realiza una búsqueda por tag. En el momento que se encuentra una coincidencia entre el tag insertado y alguno de los cues se realiza una redirección al tiempo de la escena en cuestión. Finalmente, dentro de esta funcionalidad encontramos el evento de click en alguno de los botones de los personajes. En este caso, la búsqueda se realiza a partir de encontrar el personaje en una escena aleatoria del video.

Por otra parte, encontramos el tratamiento de los gags. Los gags se tratan de escenas cortas con una finalidad humorística. Al tratarse de escenas independientes a las captadas dentro del primer VTT comentado hemos realizado un nuevo TextTrack que contiene la información (título y tags).

En este caso, la solución encontrada se basa en que en cada cambio de Cue se realiza una espera, esta espera se basa en el tiempo final del gag menos el tiempo inicial de este. Por lo tanto, al visualizar el final del gag aparece un pop-up con la valoración. Esta valoración es obligatoria para poder continuar con la reproducción.

El almacenamiento de este rating se lleva a cabo mediante variables globales dentro de JavaScript y una visualización a nivel de HTML.

Encontramos como mejoras a la práctica el tratamiento de almacenamiento del rating dentro de un servicio externo (WebServices y mostrar la información a través un HighChart) así como un tratamiento diferente al problema de la muestra del gag que no implique un desfase de tiempo debido a que pueden darse casos en los que se realiza una muestra de una valoración de un gag saltado de forma aleatoria.

# Web-responsive. Problemas y soluciones adoptadas

Des del minuto uno optamos por usar las soluciones de *bootstrap* por ser simple y elegante. La web se construyó sobre ese esquema, por lo que no fue muy complicado que fuese responsive.

# Plataformas y entornos que aceptan o no aceptan todas las funcionalidades de la práctica. Principalmente navegadores web y terminales móviles.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Chrome 72.0.3626 sobre Windows 10 Home 1809 | Firefox 66.0 sobre Windows 10 Home 1809 | Edge 44.17763.1.0 sobre Windows 10 Home 1809 | Internet Explorer 11.379 sobre Windows 10 Home 1809 | Chrome  Sobre MacOs Mojave | Firefox  Sobre MacOs Mojave | Safari  Sobre MacOs Mojave | Chrome  73.0.3683  Sobre Android Pie | Chrome  Sobre iOS | Safari  sobre iOS |
| Visor de vídeo | SI | SI | SI | SI | SI | SI | SI | SI | SI | SI |
| Subtítulos | SI | SI | SI | NO | SI | SI | SI | SI | SI | SI |
| Barra de navegación del vídeo | SI | SI | SI | SI | SI | SI | SI | SI | SI | SI |
| Personajes de esa escena destacados | SI | SI | SI | NO | SI | SI | SI | SI | SI | SI |
| Salto a una escena aleatoria de un personaje | SI | SI | SI | NO | SI | SI | SI | SI | SI | SI |
| Menú para votar un gag | SI | SI | SI | NO | SI | SI | SI | SI | SI | SI |
| Votaciones del usuario | SI | SI | SI | SI |  | SI | SI | SI | SI | SI |
| Búsqueda por tag | SI | SI | SI | NO | SI | SI | SI | SI | SI | SI |
| Reproductor de canciones | SI | SI | SI | SI | SI | SI | SI | SI | SI | SI |
| Tema oscuro | SI | NO | SI | SI | SI | NO | SI | SI | SI | SI |
| Resaltado de sección “Contacto” | SI | SI | SI | SI | SI | SI | SI | SI | SI | SI |

# ¿Qué hemos aprendido a nivel personal y de grupo?

Tras cuatro años de carrera, esta es la primera asignatura en la que interviene el elemento video y audio en una página web, por lo que se desconocía por completo su implementación. Se podía suponer que los elementos extras de cualquier video no forman parte de este, pero no sabíamos que fuese tan simple (y complejo, a veces) como añadir un track.

Se han ampliado conocimientos de funcionamiento tanto de HTML5 como de JavaScript, hecho que encontramos fundamental para salir a un mercado laboral en el que estas dos herramientas están presentes con una alta frecuencia.

La dificultad añadida de tratar con la asincronía de JavaScript y los problemas encontrados con el funcionamiento y configuración del socket nos ha aportado un conocimiento extra en estos campos.

Encontramos en esta práctica un buen ejemplo para abordar superficialmente el problema de la reproducción web e interesarse en un mundo con un gran potencial.