**Práctica: plugin**

‌

Antes de comenzar a diseñar nuestro **plugin** vamos a ver una estrategia para poder adjuntar muchos otros **plugins**.

‌

**Vamos al código**

‌

Nos dirigiremos a la función que ya hicimos llamada MediaPlayer, en ella agregaremos los plugins por el objeto de configuración.

function MediaPlayer(config) { this.media = config.el; this.plugins = config.plugins; }

Debería poder funcionar este código incluso cuando no hay plugin, No está de más dar un valor inicial.

this.plugins = config.plugins || [];

Para colocar las dos plecas verticales || usamos la combinación Alt+124.

‌

Si ya añadimos los plugins en la configuración entones ya lo podemos usar en index.js. Ya podemos parle los **plugins** por la instancia a la función que creamos.

const player = new MediaPlayer({

el: video,

plugins: []

});

‌

El primer **plugin** que vamos a agregar va a ser el que nos va a solucionar el problema del autoplay. Este **plugin** no existe, pero lo vamos a crear.

import AutoPlay from "./plugins/AutoPlay.js";

const player = new MediaPlayer({

el: video,

plugins: [new AutoPlay()]

});

‌

No estamos seguros si nuestro plugins va a recibir parámetros pero luego vemos. Vamos a crear la carpeta y el archivo inexistente que instanciamos. Cuando usamos **script** con **type movil** tenemos que ser específicos y usar la extensión .js.



En el archivo nuevo escribiremos el siguiente código:

function AutoPlay() { }

export default AutoPlay;

‌

Con esto nuestro código no tiene errores, pero tampoco tiene funcionalidades nueva. Ya lo tenemos preparado para empezar a integrar un nuevo **plugin**.

‌

En MediaPlayer vamos a necesito un tipo de inicialización.

function MediaPlayer(config) {

this.media = config.el;

this.plugins = config.plugins || [];

this.\_initPlugins();

}

MediaPlayer.prototype.\_initPlugins = function() {

this.plugins.forEach(element => {

element.run()

});

}

‌

De esta forma iteramos en cada **plugin** y lo **inicializamos** con una función llamada run(). Esta función tenemos que declararla en nuestro **plugin**.

function AutoPlay() { }

AutoPlay.prototype.run = function() {

}

export default AutoPlay;

‌

Necesitamos que esta función le de Play al video, pero tenemos que darle acceso. Para esto le pasamos una instancia del MediaPlayer, en el cual ejecutaremos las siguientes funciones.

AutoPlay.prototype.run = function (player) {

player.mute()

player.play()

}

‌

Para que la función **run** reciba **player** tenemos que pasársela en MediaPlayer usando this que representará **MediaPlayer**.

MediaPlayer.prototype.\_initPlugins = function() {

this.plugins.forEach(element => {

element.run(this)

});

}

‌

No tenemos la función mute de MediaPlayer, por ende no funcionará. Vamos a crearla.

MediaPlayer.prototype.mute = function () {

this.media.muted = true;

}

‌

Ahora crearemos un botón para que el usuario pueda mutear y desmutear cuando quiera. Para esto tenemos que crear un segundo botón en el cual llamemos por una ID ya que no será el único botón. El botón anterior también le pondremos una ID.

<button id="playPause">Play/Pause</button>

<button id="unmuteMute">Mute/Unmute</button>

‌

Llamamos correctamente a estos dos botones.

const button = document.querySelector("#playPause");

const muteUnmute = document.querySelector('#unmuteMute')

‌

Cuando le demos **click** llamará a la nueva función que crearemos para que haga **mute** y unmute.

muteUnmute.onclick = () => player.unmuteMute();

‌

El objeto player es una instancia de MediaPlayer, allí crearemos la función unmuteMute.

MediaPlayer.prototype.unmuteMute = function () {

this.media.muted ? this.media.muted = false : this.media.muted = true;

};

‌

De esta forma ya tendremos la funcionalidad de **mutear** y **desmutear**.