

# **BEAT BINDER**

## **Player Semántico Interactivo**

### **1a Documentación**



# **Beat Binder**

**Gestión y Distribución de la Información Empresarial**

Luis Miguel Vargas Durán & Mario Ventura Burgos

Universitat de les Illes Balears

Curso 2023/24

# Índice

<b>Introducción.....</b>	<b>1</b>
<b>URL Pública.....</b>	<b>1</b>
<b>Funcionalidades.....</b>	<b>2</b>
<b>Explicación técnica.....</b>	<b>3</b>
Proceso de codificación y publicación del material multimedia.....	3
Funcionalidades desarrolladas sobre los tracks (subtítulos).....	3
Funcionalidades desarrolladas del editor semántico (metadata).....	6
Metadatos y eventos del cambio de la cola.....	8
Manejo de preguntas.....	8
Selección de opciones.....	8
Funciones auxiliares.....	9
<b>Evaluación web-responsive.....</b>	<b>11</b>
En navegador web (PC).....	11
Página principal.....	11
Página de concepto.....	13
Página de catálogo.....	15
En smartphone (iPhone 11).....	16
<b>Problemas y soluciones durante la realización de la práctica.....</b>	<b>17</b>
Separación de audio y video.....	17
Creación de VTT con una estructura JSON para metadatos.....	18
<b>Opinión.....</b>	<b>18</b>
<b>Autoevaluación.....</b>	<b>19</b>
Aspectos necesarios (enunciado de la práctica).....	20
Puntos fuertes.....	21
Puntos mejorados respecto a la presentación.....	21
Puntos débiles.....	22

## **Introducción**

Beat Binder es un reproductor semántico e interactivo de vídeos, que permite a los usuarios divertirse respondiendo cuestionarios interactivos sobre sus canciones, videoclips, álbumes y artistas favoritos.

La plataforma cuenta con diferentes cuestionarios sobre diferentes canciones, sus letras, videoclips y mucho más. Esto permite a los usuarios divertirse y descubrir cuánto saben sobre sus canciones y artistas favoritos, además de aprender nuevas curiosidades. También, ofrece la posibilidad de traducir al español las canciones para poder acercarnos mucho mejor a aquello que el artista quiere transmitir.

## **URL Pública**

La dirección pública a nuestro player semántico es la siguiente:

<https://gdie2408.ltim.uib.es>

Para poder acceder y ver todos los archivos de nuestra práctica hay que saber cómo se estructuran estos archivos:

- La carpeta “css”, contiene los archivos .css con todos los estilos de la página: botones, sliders, secciones...
- La carpeta “images” contiene las imágenes que se usan en toda la página web.
- La carpeta “videos” contiene todos los videos de las distintas canciones, con sus distintas calidades de reproducción y sus diferentes extensiones para que sea soportado por más navegadores.
- La carpeta “vtt” contiene todos los archivos vtt con los que se lleva a cabo el trabajo de subtítulos y metadata. Los distintos archivos que hay son los subtítulos de las canciones que tenemos, su traducción al castellano en caso

de que la canción sea en inglés y por último, los vtt para poder controlar el modo quiz de cada canción.

- Y por último, la carpeta “js” contiene los archivos javascript para que se puedan llevar a cabo las distintas acciones de la web:
  - El ‘reproductor.js’ se llevan a cabo las funciones para que la barra de controles personalizada funcione como se espera.
  - El javascript de ‘changeVideo’ es donde se realizan las funciones para poder elegir el video y este cambie según la elección del usuario.
  - El ‘karaokeMode’ se encarga del cambio de pista del modo normal al modo karaoke o viceversa cuando el usuario clica el botón de dicho modo.
  - El javascript ‘subtitulos’ contiene las funciones para mostrar los subtítulos o desactivarlos, según quiera el usuario.
  - Por último, el js ‘quizzMode’ contiene las funciones para llevar el control del modo quiz cuando el usuario lo activa.

## Funcionalidades

Las funcionalidades con las que cuenta nuestro reproductor son las siguientes:

- Elección de videoclips de canciones
- Pausa/reproducción del videoclip
- Elección de calidad de reproducción del vídeo
- Activar/desactivar subtítulos y su traducción en caso de que sea en inglés
- Activar modo karaoke
- Activar modo quiz. Este modo funciona de tal manera que el usuario en cada pregunta que se le haga podrá elegir entre 4 opciones y podrá ver si lo ha acertado o no justo inmediatamente después de contestar.
- Ver el catálogo futuro que queremos añadir a la página
- En el catálogo futuro se puede ver la reproducción del videoclip en youtube.

## **Explicación técnica**

A continuación se explicarán de forma técnica los siguientes apartados:

### **Proceso de codificación y publicación del material multimedia**

Para optimizar la experiencia de usuario con nuestros contenidos multimedia, se ha implementado un meticuloso proceso de codificación para nuestros videos originales. Estos, capturados en alta definición a 1080p y en formato MP4, han sido transformados a varias resoluciones y formatos para adaptarse a distintas plataformas y velocidades de internet.

La codificación se ha realizado utilizando el eficiente códec H.264 para los archivos MP4. Además, para garantizar la compatibilidad con varios navegadores que prefieren el formato WEBM, hemos aplicado el códec VP9. La conversión de MP4 a WEBM se ha efectuado mediante el siguiente comando en ffmpeg desde la línea de comandos:

**`“ffmpeg -i /path/to/input.mp4 -b:v 0 -crf 30 output.webm”`**

Este procedimiento se ha repetido individualmente para cada uno de los tres videos. Se debe reemplazar input y output por los nombres correspondientes del video y /path/to/ por la ubicación específica del archivo a convertir.

En cuanto a las imágenes, no hemos realizado conversiones especiales, optando por formatos ampliamente reconocidos como PNG, JPG, JPEG y con el .ICO para el favicon. No hemos considerado necesario el uso de SVG.

### **Funcionalidades desarrolladas sobre los tracks (subtítulos)**

En este apartado se ha implementado un script en JavaScript que se encarga de manejar la funcionalidad de cambiar las pistas de subtítulos de un reproductor de video basado en eventos de clic en enlaces específicos en una página HTML. Este script hace uso de dos atributos: “data-vtt” y “data-vtt-trad” creados ambos en index.html y cuyo valor se genera al hacer click sobre una canción. A continuación, la

explicación detallada sobre cómo se usan los ficheros semánticos vtt para generar las funcionalidades de subtítulos.

En primer lugar, el script comienza escuchando el evento DOMContentLoaded, que se dispara cuando el HTML ha sido completamente cargado y analizado, sin esperar a que se carguen las imágenes y otros recursos.

Después, se busca el elemento de video en el DOM con el id videoPlayer y se le asigna a la variable video para iterar sobre los enlaces de cambio de video. En esta iteración se busca todos los elementos en el DOM que tengan la clase change-video y se asigna un *event listener* a cada uno para el evento de click.

Cuando un enlace (una canción en el apartado de elección de canción) es clicado en index.html, se obtienen los valores de los atributos **data-vtt** y **data-vtt-trad** del enlace clicado, que contienen las rutas de los archivos VTT (subtítulos) para la pista principal y la pista de traducción respectivamente. Las rutas usadas para los archivos vtt son, para todas las canciones: *public/vtt/nombreVideo.vtt*. Tras obtener las rutas de los archivos VTT, también se obtiene el idioma de la canción del atributo **data-lang** del enlace clicado. Este idioma se utilizará para configurar las etiquetas y los idiomas de las pistas de subtítulos.

Posteriormente, se crean elementos track para las pistas de subtítulos principales y de traducción. Estos elementos se configuran con los atributos *kind* (tipo de pista), *label* (etiqueta de la pista mostrada al usuario), *srlang* (source language o idioma de la pista), y *src* (ruta del archivo VTT, que es la mencionada anteriormente). La pista principal se establece como predeterminada estableciendo el valor default = true.

Finalmente, las pistas de subtítulos recién creadas se agregan como hijos del elemento de video (video) en el DOM y se carga del nuevo recurso de video mediante una llamada al método load() en el elemento de video, para cargar el nuevo recurso de video con las pistas de subtítulos asociadas.

De esta forma, se puede cambiar dinámicamente las pistas de subtítulos de un reproductor de video en una página web cuando se hace clic en enlaces específicos. Las rutas de los archivos VTT y los idiomas de las pistas de subtítulos se obtienen de los atributos de datos de los enlaces clicados, y se configuran las pistas de subtítulos en consecuencia.

Cabe mencionar que solo existen traducciones de las canciones en inglés, es decir, que si se detecta que una canción está en inglés, tendrá dos ficheros vtt asociados (original y traducción), pero si la canción está en español, esta no dispondrá de traducción, haciendo que el atributo data-vtt-trad definido en index.html sea null.

A continuación el código del script **subtitulos.js**

```
document.addEventListener('DOMContentLoaded', () => {
    // Se obtiene el reproductor por su id
    const video = document.getElementById('videoPlayer');

    document.querySelectorAll('.change-video').forEach(link => {
        link.addEventListener('click', function (e) {
            const newVttSrc = this.getAttribute('data-vtt');
            const newVttSrcTrad = this.getAttribute('data-vtt-trad');
            // Idioma de la canción
            const songLanguage = this.getAttribute('data-lang');
            let subtitleLang = null;

            // Comprobación del idioma
            if (songLanguage == 'es') {
                subtitleLang = 'Español';

            } else if (songLanguage == 'en') {
                subtitleLang = 'Inglés';
            }

            // Añade la pista de subtítulos principal
            if (newVttSrc) {
                const track = document.createElement('track');
                track.kind = 'subtitles';
                // Usa la etiqueta basada en el idioma de la canción
                track.label = subtitleLang;
                track.srclang = songLanguage;
                track.src = newVttSrc;
            }
        });
    });
}
```

```

        track.default = true;
        video.appendChild(track);
    }

    if (newVttSrcTrad) {
        const trackTrad = document.createElement('track');
        trackTrad.kind = 'subtitles';
        // Subtítulos en español para canciones en inglés
        trackTrad.label = 'Español';
        trackTrad srclang = 'es';
        trackTrad.src = newVttSrcTrad;
        video.appendChild(trackTrad);
    }
    video.load(); // Carga el nuevo recurso
);
);
);
);
}

```

## Funcionalidades desarrolladas del editor semántico (metadata)

Este código JavaScript se ejecuta una vez que el documento HTML ha sido completamente cargado y parseado, como indica el evento `DOMContentLoaded`. Está diseñado para gestionar el reproductor multimedia interactivo que tiene la capacidad de presentar el cuestionario (modo quizz) basado en los metadatos del vídeo.

El script comienza estableciendo una serie de constantes y variables que hacen referencia a elementos del DOM, como el reproductor de vídeo ('videoPlayer'), el reproductor de audio ('audioPlayer'), los botones de opciones de quizz y los contenedores de texto para las preguntas y mensajes al usuario.

```

document.addEventListener('DOMContentLoaded', () => {
    // VARIABLES Y CONSTANTES
    const videoPlayer = document.getElementById('videoPlayer');
    const audioPlayer = document.getElementById('audioPlayer');
    const quizzModeBtn = document.getElementById('quizzModeBtn');
    var questionTitle = document.getElementById('questionTitle');
    var questionText = document.getElementById('questionText');
    var messageText = document.getElementById('messageText');
    var option1 = document.getElementById('option0');
    var option2 = document.getElementById('option1');
    var option3 = document.getElementById('option2');
    var option4 = document.getElementById('option3');
    var puntosTotales = 0;
    var modoActual = 0;
    var numPregunta = -1;
    var idx = 0;
    var preguntasVideo;
    var stringVideo;
    var firstTime = true;

```

La función ‘quizzMode’ es un controlador de eventos que se activa al hacer clic en el botón de modo quiz (‘quizzModeBtn’). La funcionalidad se centra en alternar entre el modo de visualización normal del video y el modo quiz. Al activarse, restablece varias variables a sus estados iniciales para garantizar una transición limpia entre modos.

```

// Evento asociado al click en el botón de cambio de modo
quizzModeBtn.addEventListener('click', () => {
    resetVars();
    // Deshabilitar botones
    disableOptions();
    // Ir a quizz mode
    quizzMode();
})

```

El modo quizz utiliza los metadatos de las pistas de texto (‘textTracks’) del elemento de vídeo para sincronizar las preguntas del quiz con momentos específicos del video. A continuación, la explicación acerca de cómo se maneja.

## Metadatos y eventos del cambio de la cola

El código utiliza un track de metadatos para insertar preguntas del quiz en el video. El evento 'cuechange' se dispara cuando el tiempo actual del video entra en el rango de tiempo de una "cue" (una señal o marca) dentro de esta pista. Este mecanismo asegura que las preguntas se presenten en los tiempos correctos.

```
quizzTrack.addEventListener('cuechange', function () {
    // Se establece el título de la pregunta
    enableOptions();
    let cue = this.activeCues[0];
    console.log(cue);
    if (cue){
        mediaPlayer.pause();
        audioPlayer.pause();
        questionTitle.textContent = "Pregunta #" + (numPregunta += 1);
        preguntasVideo = JSON.parse(cue.text);
        questionText.textContent = preguntasVideo.pregunta;
        option1.textContent = preguntasVideo.respuestas[0];
        option2.textContent = preguntasVideo.respuestas[1];
        option3.textContent = preguntasVideo.respuestas[2];
        option4.textContent = preguntasVideo.respuestas[3];
        option1.addEventListener('click', () => opcionSeleccionada(0, preguntasVideo.correcta));
        option2.addEventListener('click', () => opcionSeleccionada(1, preguntasVideo.correcta));
        option3.addEventListener('click', () => opcionSeleccionada(2, preguntasVideo.correcta));
        option4.addEventListener('click', () => opcionSeleccionada(3, preguntasVideo.correcta));
    }
});
```

## Manejo de preguntas

Cuando ocurre un 'cuechange', se pausa el video y el audio, y se muestra la pregunta asociada en la pantalla. Los botones de opción se habilitan para que el usuario pueda seleccionar una respuesta.

## Selección de opciones

Los botones de opción tienen event handlers que se activan cuando se seleccionan. La función 'opcionSeleccionada' verifica si la opción elegida es correcta, actualiza la interfaz de usuario con un mensaje apropiado y cambia el estilo de los botones para reflejar la corrección de la respuesta.

```

function opcionSeleccionada(idOpcionSeleccionada, correcta) {
    // deshabilitamos las opciones
    disableOptions();
    // Generamos un string con el valor de la opción correcta
    var idCorrecto = "option" + correcta;
    // Se comprueba si el pulsado y el correcto coinciden
    if (idOpcionSeleccionada == correcta) {
        // messageText con mensaje que avisa al usuario
        messageText.textContent = "¡Correcto!";
    } else {
        // messageText con mensaje que avisa al usuario
        messageText.textContent = "Has fallado...";
    }

    // Esperar 3 segundos antes de reanudar el vídeo y mostrar la siguiente
    setTimeout(() => {
        actualizarInterfaz();
        videoPlayer.play(); // Reanudar el vídeo
        audioPlayer.play();
    }, 3000);

    // Cambiar colores de los botones según si es correcto o incorrecto
    document.getElementById(idCorrecto).classList.add("correctStyle");
    // Los botones incorrectos aparecerán de color rojo
    for (let i = 0; i < 4; i++) {
        var idBoton = "option" + i;
        // Verificar si el botón actual no es el correcto
        if (idBoton !== idCorrecto) {
            // Marcar el botón actual como incorrecto
            document.getElementById(idBoton).classList.add("wrongStyle");
        }
    }
}

```

## Funciones auxiliares

El script incluye funciones auxiliares como ‘resetVars’, ‘disableOptions’, ‘enableOptions’, y ‘actualizarInterfaz’ que se utilizan para controlar el flujo del quizz, como restablecer la interfaz para la siguiente pregunta o deshabilitar/habilitar los botones de opción durante la evaluación de las respuestas.

```
function resetVars() {
    // Resetea los parametros necesarios para que el modo quizz se active
    firstTime = true;
    messageText.textContent = "";
    document.getElementById('continuePlaying').style.display = "none";
    puntosTotales = 0;
    lastQuestionTime = -1;
    idx = 0;
    numPregunta = -1;
    document.getElementById('option0').className = "buttonOp1";
    document.getElementById('option1').className = "buttonOp2";
    document.getElementById('option2').className = "buttonOp3";
    document.getElementById('option3').className = "buttonOp4";
    videoPlayer.currentTime = 0;
    audioPlayer.currentTime = 0;
}
```

```
// Inhabilita las funciones de los botones de opciones de respuesta
function disableOptions() {
    var options = [option1, option2, option3, option4];
    options.forEach(option => {
        option.disabled = true;
    });
}

// Habilita las funciones de los botones de opciones de respuesta
function enableOptions() {
    var options = [option1, option2, option3, option4];
    options.forEach(option => {
        option.disabled = false;
    });
}
```

```

function actualizarInterfaz() {
    // Restablecer el título de la pregunta y el texto de la pregunta para la siguiente visualización
    questionTitle.textContent = "PlayQuizz";
    questionText.textContent = "Mira el video hasta que aparezca la siguiente pregunta";
    option1.textContent = "Opción 1";
    option2.textContent = "Opción 2";
    option3.textContent = "Opción 3";
    option4.textContent = "Opción 4";
    document.getElementById('option0').className = "buttonOp1";
    document.getElementById('option1').className = "buttonOp2";
    document.getElementById('option2').className = "buttonOp3";
    document.getElementById('option3').className = "buttonOp4";
    messageText.textContent = "";
}

```

## Evaluación web-responsive

Para evaluar si Beat Binder es web-responsive, se abrirá en una ventana en el navegador Google Chrome y se mostrará el aspecto con diferentes tamaños de ventana. Posteriormente, se abrirá también usando un smartphone, que en este caso, es un iPhone 11.

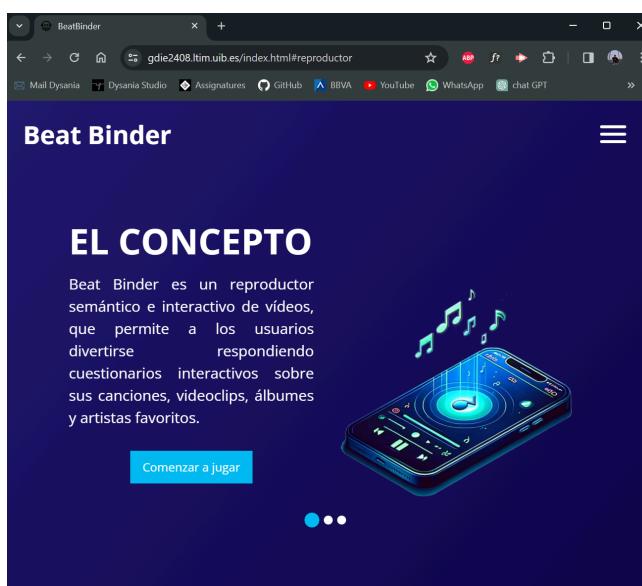
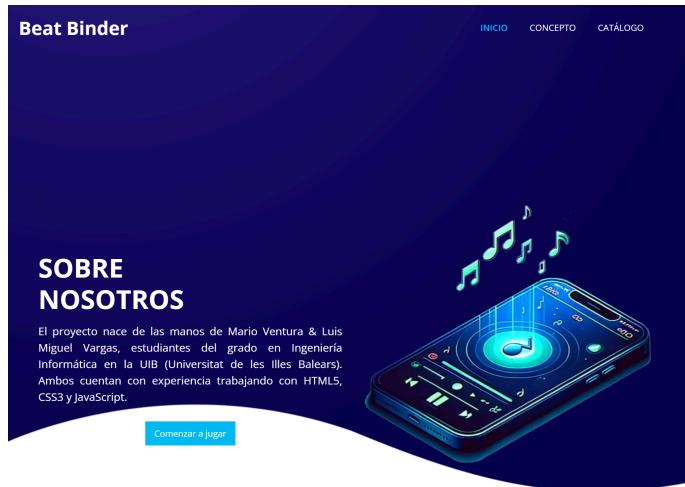
### En navegador web (PC)

#### Página principal

Al abrir Beat Binder (index.html) con pantalla completa, se ve lo siguiente:



Al reducir el tamaño de la ventana a la mitad se puede ver que el formato se mantiene, y la imagen azul que dibuja una “ola” en el fondo, también se adapta al tamaño de la ventana.



Por otro lado, si se hace la ventana todavía más pequeña, para evitar estilos indeseados al estrechar en exceso la imagen, esta desaparece y se pasa a un fondo sólido que mantiene la estética de la página.

Además, los tres elementos que aparecen en la barra de navegación (Inicio, Concepto y Catálogo) dejarán de aparecer y, en su lugar, aparecerá el ícono que se ve en la esquina superior derecha. Este ícono despliega esas tres opciones a modo de columna vertical al ser pulsado, y además usa una transición para hacerlo, haciendo que Beat Binder sea más user-friendly.

# Beat Binder

X

INICIO

CONCEPTO

CATÁLOGO

Esto se hace gracias a que la zona que ocupa la imagen tiene unas propiedades de tamaño mínimo y máximo, haciendo que si el tamaño de la ventana es inferior al tamaño mínimo establecido, se recurra a un color de fondo sólido.

## Página de concepto

En el apartado de concepto que presenta la página, se describe qué es Beat Binder. para ello se usa un slider idéntico al de la página de inicio, igualmente responsive ya que es el mismo código pero con texto e imágenes distintas. Para no extender en exceso el documento, no se añadirán más imágenes.



Para no extender en exceso el documento, no se añadirán más imágenes, ya que es igual de responsive que el slider del apartado anterior al ser el mismo código de slider con las mismas funcionalidades y propiedades.

En este mismo apartado, se muestran las canciones disponibles actualmente junto con un botón que llama a la acción al usuario para invitarlo a jugar y probar Beat Binder. Este apartado también es web-responsive, como se puede ver en las imágenes siguientes (ventana completa y ventana de tamaño reducido, respectivamente):

**Canciones Disponibles**  
Actualmente podrás elegir entre las siguientes canciones:



Amorfoda



Beautiful Girls



La Ultima

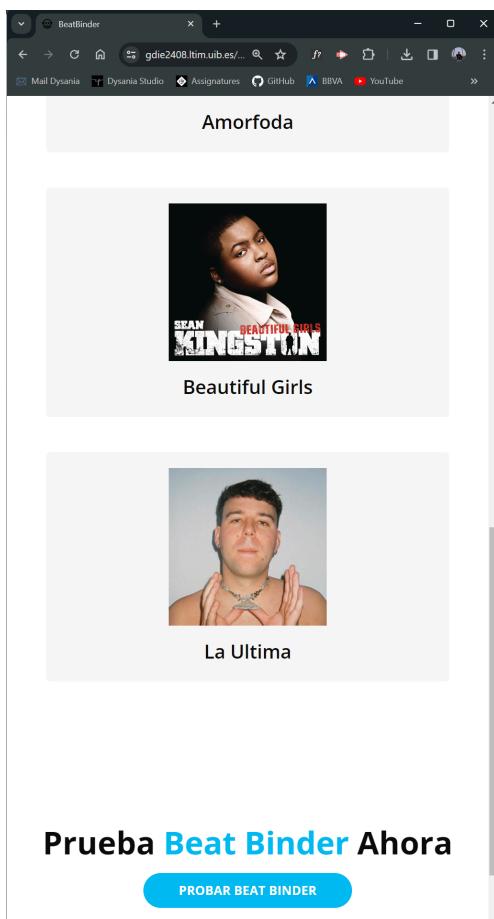
  

**Prueba Beat Binder Ahora**

[PROBAR BEAT BINDER](#)

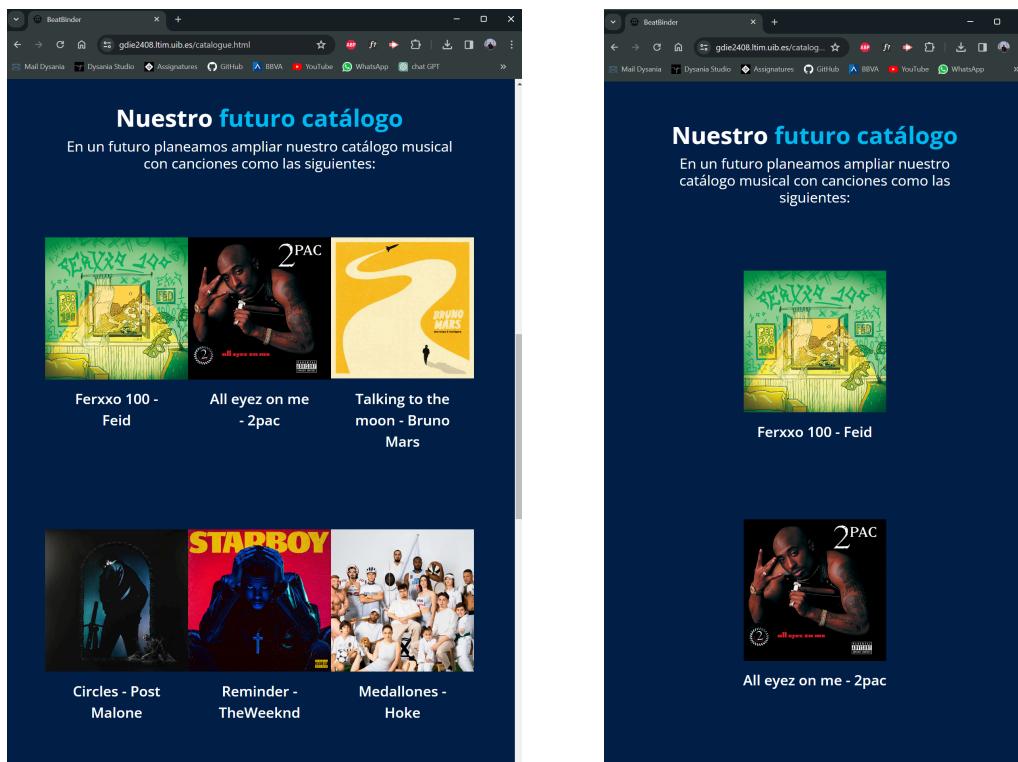
---

© 2024 All Rights Reserved By UIB



## Página de catálogo

Al abrir esta página nos encontramos con el mismo inicio que en los otros dos apartados, pero en este caso el inicio es estático, no un slider. Es igualmente responsive ya que es el mismo código pero sin hacerlo slider, es decir, es como uno de los 3 apartados responsive que tienen los sliders mostrados. Posteriormente, hay una lista de canciones, que siguen el mismo patrón de diseño que el apartado de selección de canción en la página de inicio. A continuación, las imágenes que muestran que este apartado también es responsive.

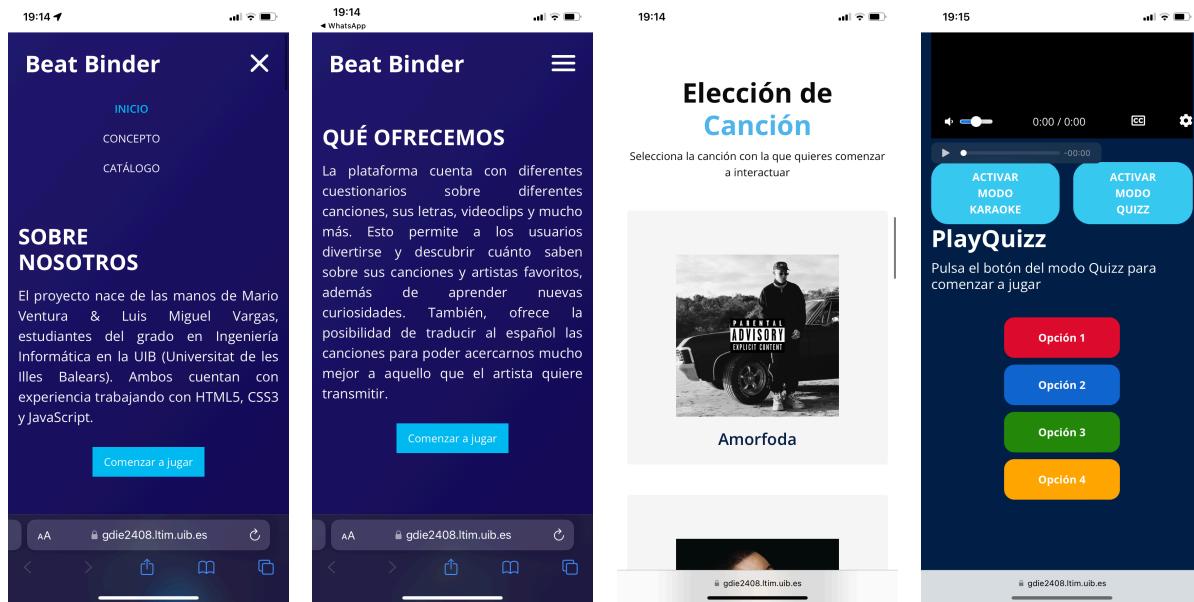


En el próximo apartado se abrirán los mismos apartados de la página en un smartphone para ver si el diseño se adapta también a una pantalla con orientación vertical y dimensiones distintas. De esta forma se podrá hacer una evaluación realista acerca de si Beat Binder es responsive.

## En smartphone (iPhone 11)

A continuación se mostrarán capturas de como se ve Beat Binder en un smartphone. para no extender el informe en exceso, simplemente se mostrará las imágenes correspondientes a diferentes zonas de la página, mostrando que es responsive ya que su diseño se adapta al dispositivo en concreto.

*Inicio con navbar, inicio sin navbar, elección de canción y modo quizz*



## Autores, futuro catálogo, llamada a la acción y footer

The image displays four screenshots of a mobile application interface, likely a prototype or final product for a music catalog. The first two screenshots show profiles of two students: Mario Ventura and Luis Miguel Vargas. The third screenshot shows a video player with a thumbnail for 'Ferxxo 100 - Feid' and a play button labeled 'La Ultima'. The fourth screenshot shows a footer section with contact information, social media links, and a newsletter sign-up form.

- Los Autores:** Shows profile pictures and names of two students: MARIO VENTURA and LUIS MIGUEL VARGAS. Below their names are their academic details: MARIO VENTURA is an Informatics Engineering student studying Systems of Information (Sistemas de Información) using HTML5, CSS3, and JavaScript; LUIS MIGUEL VARGAS is also an Informatics Engineering student studying Technologies of Information (Tecnologías de la Información) using HTML5, CSS3, and VTTs. At the bottom of each profile is a URL: [gdie2408.itim.uib.es](http://gdie2408.itim.uib.es).
- Nuestro futuro catálogo:** Shows a title 'Nuestro futuro catálogo' and a text: 'En un futuro planeamos ampliar nuestro catálogo musical con canciones como las siguientes:' followed by a thumbnail image of a song cover for 'Ferxxo 100 - Feid'. Below the thumbnail is another URL: [gdie2408.itim.uib.es](http://gdie2408.itim.uib.es).
- Video Player:** Shows a thumbnail for 'La Ultima' by Ferxxo 100 - Feid' and a play button. Below the play button is a URL: [gdie2408.itim.uib.es](http://gdie2408.itim.uib.es).
- Footer:** Contains sections for 'Dirección', 'Info', 'Links', and 'Suscríbete'. 'Dirección' includes an address, phone number, and email. 'Info' provides project details. 'Links' includes links to 'Inicio', 'Concepto', and 'Catalogo'. 'Suscríbete' includes a newsletter sign-up form with fields for 'Introduce tu email' and a 'Suscríbete' button. At the bottom is a URL: [gdie2408.itim.uib.es](http://gdie2408.itim.uib.es).

## Problemas y soluciones durante la realización de la práctica

Durante la realización de la práctica, nos enfrentamos a varios desafíos técnicos, entre ellos la separación del audio y el video y la generación de archivos VTT con una estructura JSON para manejar metadatos. A continuación, se detalla cada uno de estos problemas y las soluciones implementadas para superarlos:

### Separación de audio y video

La separación del audio y el video nos ha presentado desafíos significativos, especialmente en lo que respecta a la sincronización entre ambos, lo que afectaba la experiencia del usuario al consumir el contenido multimedia.

Para resolver este problema, se desarrollaron funciones de envoltura en JavaScript que controlaban la reproducción de audio y video de manera sincronizada. Así, cualquier acción en el reproductor de video (como reproducir, pausar o adelantar el video) se reflejaba simultáneamente en el reproductor de audio.

## **Creación de VTT con una estructura JSON para metadatos**

La necesidad de incorporar metadatos complejos, como preguntas de quizz y sus respectivas respuestas, en los archivos VTT, requería un formato estructurado y flexible.

Se optó por una estructura JSON dentro de los archivos VTT, lo que permitió un manejo más avanzado de los metadatos. Se crearon herramientas que permitían a los creadores de contenido insertar fácilmente los datos necesarios en un formato JSON, que luego se convierte y queda contenido en archivos VTT. Esta solución no solo proporcionó la estructura requerida para los metadatos, sino que también facilitó su mantenimiento y actualización.

## **Opinión**

Tras la realización del trabajo, se ha podido apreciar la relevancia de, entre otros, los siguientes aspectos:

1. La importancia de los ficheros VTT y su facilidad de uso se destaca como un punto crucial, facilitando tanto la implementación como la manipulación de subtítulos y otros textos sincronizados con el vídeo. Esta accesibilidad de los ficheros VTT ha permitido una integración fluida y eficaz, reforzando la interactividad y la accesibilidad del contenido multimedia utilizado.
2. La versatilidad de los archivos VTT también ha desempeñado un papel fundamental, extendiéndose más allá de la simple provisión de subtítulos. El uso innovador de estos ficheros para el modo quizz ha demostrado su flexibilidad, permitiendo que se utilicen de maneras creativas para aumentar el engagement y ofrecer una experiencia de usuario enriquecedora y dinámica.
3. Además, la gestión adecuada del tamaño de los archivos multimedia se ha revelado como un aspecto crítico. Al optimizar los vídeos y otros contenidos para que mantengan una calidad aceptable con un tamaño de archivo

reducido, se ha evitado que la página se vea afectada por tiempos de carga lentos. Esto no solo mejora la experiencia del usuario sino que también favorece el desempeño general del sitio, garantizando que los recursos multimedia contribuyan positivamente al valor y funcionalidad del sitio web sin comprometer la velocidad o eficiencia.

Estos aspectos son interesantes ya que pese a ser relevantes, es la primera vez que se tratan en los 4 años del grado. La práctica plantea un ejercicio completo de implementación de un producto que requiere interacción del usuario. Esta interacción usa ficheros semánticos en diferentes formatos para diferentes funcionalidades y ha sido de gran utilidad para conocer cómo se debe trabajar a nivel de código con archivos multimedia, qué características tienen y qué utilidades presentan.

Esta práctica, por tanto, ha ayudado a los autores a asimilar un conocimiento necesario y de gran utilidad a través de diferentes desafíos y soluciones presentadas a ellos.

## **Autoevaluación**

Para poder evaluar el trabajo realizado con cierta objetividad se valorarán diferentes aspectos, teniendo en cuenta los establecidos en el enunciado, los puntos mejorados respecto al día de la presentación en clase, los puntos fuertes (cosas bien hechas y requisitos cumplidos) y los puntos débiles (aspectos mejorables o requisitos no cumplidos). De esta forma se podrá justificar la nota puesta.

## Aspectos necesarios (enunciado de la práctica)

ASPECTO	NOTA (1-10) + JUSTIFICACIÓN
Creació d'un servidor HTTP al servidor virtual.	10 (hecho).
Reproductor de vídeo en una aplicació web.	10 (hecho).
Ús de VideoTextTrack per a subtítols (captions) i metadades (metadata).	10 (hecho).
Adequació entre la funcionalitat i el cas d'ús triat.	10 (correcto)
Qualitat dels elements multimèdia.	7 (buena, pero mejorable).
Selecció manual de diferents qualitats de vídeo, streams d'àudio i subtítols.	7 (hay pistas de audio y subtítulos con ficheros vtt, pero no hay diferentes calidades).
Estil visual de l'aplicació.	10.
Disseny responsive.	10.
Ús de més tracks multimèdia, com pistes d'audio o altres pistes de text.	8 (se utilizan pistas de audio para el control del modo Karaoke)
Possibilitat de triar entre més d'un vídeo.	10 (existen 3 vídeos sobre los que interactuar con ficheros semánticos vtt en modo Karaoke y modo Quizz).
Reproductor de vídeo avançat, amb controls interactius a través de metadades semàntiques.	7 (hay funcionalidad de subtítulos y preguntas en el modo Quizz hechas con vtt, pero podrían añadirse más).

Ús d'APIs externes relacionades amb les metadades semàntiques.	No Implementado.
Inclusió d'un editor per les metadades semàntiques.	No Implementado.
Altres aspectes relacionats amb la inclusió d'interactivitat basada en la semàntica del contingut multimèdia.	No Implementado.

## Puntos fuertes

1. **Entorno / atmósfera creada.** Beat Binder tiene una interfaz visual y una identidad de marca (paleta de colores, concepto, etc.) que añaden valor a la plataforma más allá de sus funcionalidades. Se considera que el entorno creado es muy bueno.
2. **Diseño responsive.** La página es web-responsive y mantiene su estilo en pantallas y dispositivos de diferentes dimensiones.

## Puntos mejorados respecto a la presentación

1. **Modo quizz con ficheros vtt.** Beat Binder permite reproducir varios videos e interactuar en todos ellos gracias a ficheros semánticos. El modo Quizz está presente en todas las canciones, teniendo para cada video un total de 3 preguntas con 4 respuestas cada una. Estas preguntas están hechas en archivos vtt en formato JSON. Además, el modo karaoke y los subtítulos también están disponibles para todos los videos, y este último también funciona con ficheros vtt. Cabe recalcar que, en la presentación esto no se realizaba haciendo uso de los

2. **Separación y sincronización de audio y vídeo.** En la presentación hecha en clase, se observó que el diseño del modo karaoke no era el más óptimo ya que funcionaba con otro track de vídeo, donde este vídeo solo tenía la instrumental de la canción. De esta forma, se necesitaban dos vídeos idénticos para cada canción, uno para el modo normal y otro para el modo karaoke, con la única distinción del audio usado. Se dijo que esto no era óptimo por el peso que suponían los vídeos, y se ha corregido de forma que ahora, para cada canción, se use solo un track de vídeo y dos de audio en función del modo de reproducción.
3. **Reducción de espacio en el servidor.** Como consecuencia del punto anterior, se reduce el espacio usado en el servidor al no usar dos tracks de vídeo por cada canción.

## Puntos débiles

1. No se dispone de diferentes calidades de vídeo.
2. No se usan API's o librerías externas.
3. No se implementa un editor de los datos semánticos.

Teniendo todos estos aspectos en cuenta, y evaluando la relevancia que se considera que tiene cada uno de ellos, **se estima una nota final de entre 6,5 y 8,0**. Se cree que esta nota es acorde al trabajo presentado ya que se cumplen con los requisitos obligatorios y se implementan algunos opcionales. Aun así, el trabajo no es excelente ya que hay cierto margen factible de mejora