

Difusión de un DJ

UOC

Eric Atencia Montero

TFG

Gestión y publicación de
contenidos

Nombre Tutor/a de TF

Carlos Casado Martínez

**Profesor/a responsable de
la asignatura**

Daniel Julià Lundgren

Fecha Entrega

17/06/2023



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-

SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Ficha del trabajo final

Título del trabajo:	<i>Difusión de un DJ</i>
Nombre del autor:	<i>Eric Atencia Montero</i>
Nombre del consultor/a:	<i>Daniel Julià Lundgren</i>
Nombre del PRA:	<i>Carlos Casado Martínez</i>
Fecha de entrega (mm/aaaa):	<i>06/2023</i>
Titulación o programa:	Grado Multimedia
Área del Trabajo Final:	<i>Gestión y publicación de contenidos</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Podcasting, Wordpress, LAMP, Apache, MySQL, PHP, phpMyAdmin, Erik Atia, DJ, FTP, Server, Ubuntu, Amazon, AWS</i>
Abstract	
<p>La finalidad de este proyecto es profesionalizar mi imagen como DJ, creando una página web completa donde poder contar mi historia, poder hacer públicas mis sesiones de música, dar un lugar donde los clientes potenciales me puedan contactar, y además de todo eso quiero aprovechar para el control de todo esto no cedérselo a una empresa de terceros, si no que quiero llevarlo a cabo con un servidor del que yo disponga del control.</p> <p>Por otro lado, a día de hoy hay tengo un podcast donde subo sesiones de música alojado en una plataforma llamada iVoox, quiero difundir este contenido a la mayoría de plataformas de podcasting, y que además cuando suba el contenido en la principal, automáticamente se publique en el resto. Además, profesionalizaré la imagen del programa haciendo covers nuevos.</p>	



Índice

Ficha del trabajo final	3
Índice de figuras	7
1. Introducción	9
1.1. Contexto y justificación del trabajo	9
1.2. Objetivos del trabajo.....	11
1.3. Impacto en sostenibilidad, ético-social y de diversidad	12
1.4. Enfoque y método seguido.....	13
1.5. Planificación del trabajo.....	14
1.6. Breve sumario de productos obtenidos	14
1.7. Breve descripción de los otros capítulos de la memoria.....	15
2. Materiales y métodos	16
2.1. Máquina virtual.....	16
2.2. Album Covers.....	18
2.3. Difusión Podcast	19
2.4. Wordpress	19
2.5. Web RSS (HTML + CSS)	21
2.6. NOIP	22
2.7. SSL For Free.....	23
3. Resultados	24
3.1. Servidor LAMP + Wordpress.....	24
3.2. Album Covers.....	24
3.3. Difusión Podcast	25
3.4. Web artista (Wordpress)	28
3.5. Web RRSS (HTML + CSS)	31
Problemas	32
Conclusiones y trabajos futuros	33
4. Glosario	35
5. Bibliografía	36
6. Anexos	38
6.1. Creación de la máquina virtual	38
6.2. Conexión con el servidor	41
6.3. Actualización del servidor	41
6.4. Instalación y configuración LAMP.....	41
6.5. Instalación Wordpress.....	45

6.6.	Difusión del Podcast.....	47
6.7.	Certificado SSL	49
6.8.	FTP.....	49
6.9.	Creación Web.....	51
6.9.1.	Home.....	52
6.9.2.	Social	54
6.9.3.	About Me	55
6.9.4.	Podcast.....	56
6.9.5.	Contact.....	57
6.10.	Plugins	58
6.10.1.	Smash Ballon Instagram Feed	58
6.10.2.	WP Mail SMTP.....	59

Índice de figuras

FIGURA 1: DIAGRAMA DE GANTT	14
FIGURA 2: PLATILLA FREEPIK	18
FIGURA 3: IFRAME IVOOX.....	20
FIGURA 4: IFRAME APPLE PODCASTS.....	21
FIGURA 5: CÓDIGO HTML	21
FIGURA 6: CÓDIGO CSS.....	22
FIGURA 7: NOIP	22
FIGURA 8: SSL FOR FREE.....	23
FIGURA 9: PORTADA PODCAST	24
FIGURA 10: EPISODIOS PODCAST.....	25
FIGURA 11: APPLE PODCASTS	25
FIGURA 12: AMAZON MUSIC.....	26
FIGURA 13: GOOGLE PODCASTS	26
FIGURA 14: CASTBOX.....	27
FIGURA 15: HOME	28
FIGURA 16: SOCIAL.....	29
FIGURA 17: ABOUT ME	29
FIGURA 18: PODCAST	30
FIGURA 19: CONTACT	30
FIGURA 20: RRSS.....	31
FIGURA 21: CREACIÓN MÁQUINA VIRTUAL 1	38
FIGURA 22: CREACIÓN MÁQUINA VIRTUAL 2.....	38
FIGURA 23: ARCHIVO .PEM.....	39
FIGURA 24: MÁQUINA CREADA	39
FIGURA 25: DETALLES MÁQUINA.....	40
FIGURA 26: CONEXIÓN SSH A LA VM.....	41
FIGURA 27: ACTUALIZACIÓN REPOSITORIOS.....	41
FIGURA 28: ACTUALIZACIÓN DE PAQUETES	41
FIGURA 29: INSTALACIÓN APACHE	41
FIGURA 30: FIREWALL APACHE.....	41
FIGURA 31: INSTALACIÓN PAQUETE REDES	41
FIGURA 32: APACHE FUNCIONANDO	42
FIGURA 33: INSTALACIÓN MYSQL SERVER	42
FIGURA 34: INSTALACIÓN PHP	42
FIGURA 35: INSTALACIÓN PHPMYADMIN	42
FIGURA 36: PHPMYADMIN FUNCIONANDO.....	43
FIGURA 37: CREACIÓN USUARIO MYSQL SERVER.....	43
FIGURA 38: PERMISOS MYSQL SERVER	43
FIGURA 39: APLICAR PERMISOS	43
FIGURA 40: INICIO DE PHPMYADMIN	44
FIGURA 41: CREACIÓN BBDD.....	45
FIGURA 42: DESCARGA DE WORDPRESS	45
FIGURA 43: BIENVENIDA DE WORDPRESS	46
FIGURA 44: INICIO DE WORDPRESS	46
FIGURA 45: FEED RSS.....	47
FIGURA 46: APPLE PODCASTS CONNECT.....	48
FIGURA 47: CERTIFICADOS.....	49
FIGURA 48: FICHERO CONFIGURACIÓN APACHE	49

FIGURA 49: INSTALACIÓN VSFTPD.....	49
FIGURA 50: CONFIGURACIÓN VSFTP.....	50
FIGURA 51: SECCIÓN	51
FIGURA 52: FILA	51
FIGURA 53: MODULO	51
FIGURA 54: HOME 1	52
FIGURA 55: HOME 2	53
FIGURA 56: HOME 3	53
FIGURA 57: SOCIAL 1.....	54
FIGURA 58: ABOUT ME 1	55
FIGURA 59: PODCAST 1	56
FIGURA 60: CONTACT 1	57
FIGURA 61: SMASH BALLON INSTAGRAM FEED	58
FIGURA 62: FEED 1	58
FIGURA 63: FEED 2	59
FIGURA 64: WP MAIL SMTP.....	59

1. Introducción

1.1. Contexto y justificación del trabajo

Para contextualizar este TFG, creo que es muy importante incluir la parte motivacional, y es que cuando finalice este proyecto, el resultado final va a ser útil para mí y mi proyecto como DJ. Todo lo que voy a relatar a continuación sobre los puntos que quiero conseguir con este trabajo son cosas que siempre había querido realizar pero que jamás me había puesto a hacerlas. Creo que ahora que tengo los conocimientos para poder llevarlas a cabo es un buen momento para realizarlo.

Las oportunidades que nos proporciona este proyecto son bastante definidas, al final lo que queremos conseguir es darnos más visibilidad como artista y sobre todo una imagen más profesional, dentro de lo que aporta conseguir todo, entre otras cosas, lo más importante es conseguir más actuaciones como DJ, esto viene promovido por una imagen más profesional y hacer mucho más visibles las sesiones de música.

Es una industria muy complicada, competimos contra un gran número de personas que quieren abrirse un hueco y destacar por encima del resto, son muy pocos los que lo consiguen. Como competidores directos, tenemos todo aquel podcast dedicado a la música, aunque más concretamente al de música electrónica, que es el subgénero sobre el que trabajo yo. Sin ir más lejos, iVoox, tiene según sus informas, más de 1 millón de podcast, y el mío a día de hoy está en estas posiciones. Es decir, el número de podcast contra los que debemos competir es considerable. Es por eso que debemos hacer cosas distintas que nos hagan destacar por encima del resto.

Como herramientas para desarrollar este proyecto, utilizaremos varias.

Para la parte del servidor web y de la propia web, trabajaremos en un entorno de máquina virtual Linux, a priori será un Ubuntu Server, ya que estoy muy familiarizado con él. Dentro instalaremos un paquete LAMP (Linux Apache MySQL PHP), el cual será el encargado de correr nuestra web.

Como CMS para la creación de la web utilizaré Wordpress, el cual potenciaremos con diversos plugins, plantillas, etc.

He trabajado anteriormente con Wordpress y pese a ser una herramienta bastante “user friendly” a veces es demasiado cerrada en según qué aspectos, dependiendo que plantilla utilicemos puede variar más o menos. Por otro lado, uno de los retos que quiero conseguir es integrar la página con Instagram y con Twitter, para mostrar las últimas publicaciones, lo he intentado en alguna ocasión y no ha sido posible.

Además, crearé una pequeña subweb hecha únicamente a código en HTML y CSS para utilizar como multienlace en las redes sociales.

Todo el tema de los covers y todo lo que tenga que diseñarse, se desarrollará con el pack Adobe CC, principalmente con Adobe Illustrator y Adobe Photoshop.

1.2. Objetivos del trabajo

Con este TFG quiero tratar de palear un problema de visibilidad que tiene mi faceta de DJ en general, es por ello que quiero conseguir profesionalizar el proyecto. De ahí el desarrollo de la web y querer ampliar mucho más la visibilidad del podcast, queriendo expandirlo a multitud de plataformas de podcasting. Con todo esto, el objetivo es claro, es conseguir llegar a un número de usuarios mayor que el que llevo actualmente. Para conseguir llegar a todo esto, esta es la serie de productos que vamos a tener en producción:

- Servidor LAMP + FTP: Un servidor web totalmente funcional, además de FTP para poder subir contenido de manera rápida y sencilla.
- Web de artista completa hecha en Wordpress: Accesible desde <https://erikatia.sytes.net>
- Web RRSS hecha en HTML y CSS: Su finalidad es la de la difusión en redes sociales cuando se lancen episodios nuevos en el podcast, puesto que esta tendrá un enlace a todas las plataformas donde estará disponible el programa. Esta, estará accesible desde: <https://erikatia.sytes.net/rrss>
- Covers personalizados para cada episodio y para el programa.

Esta es la lista de recursos digitales que conseguimos con este proyecto y con los que vamos a llegar a nuestro objetivo.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

En apartados de sostenibilidad considero que el proyecto no tiene ni aspectos positivos ni negativos prácticamente, ya que se trata de un proyecto meramente de software, no requiere de ningún tipo de producción física. El único coste que puede tener es, el coste energético que puede tener de media un servidor web como el que debemos montar. De media un servidor web consume unos 200 vatios/h, al año acabarían siendo unos 1800 KWh aproximadamente. Si nos vamos al tema de las emisiones CO2, depende mucho del país donde lo alojemos y del sistema de energías que se utilicen, si son renovables, nuclear, etc. En el caso del modelo europeo deberíamos estar enviando alrededor de unos 525Kg de CO2 aproximadamente. Tanto el consumo eléctrico como el envío de emisiones CO2 es aproximado y orientativo, depende de múltiples factores como los que ya hemos comentado, además de la carga de trabajo del propio servidor, ya que contra menos carga tenga, menos consumirá y a su vez menos emisiones enviará.

La ética es un tema complicado, la web en si opino que no tiene impactos positivos ni negativos en cuanto temas ético-sociales. Debido a que todo el material que se utilizaría es propio, de creación propia y que no implica a nadie más, pero, debemos entrar también en un tema complicado que es el uso de la música en el podcast, pese a que no es el tema de este TFG, vamos a crear este trabajo un tanto alrededor de este podcast. Debido a eso, debemos tener en cuenta todo el tema legal de los derechos de autor que incluyen las canciones que suenan. Esto se solventa de manera muy sencilla, debemos acudir a plataformas que la mayoría de plataformas relacionadas con este podcast cuentan con licencias para poder emitir música con copyright de igual modo que un club nocturno también debe tenerla.

Ahora, planteemos el hecho de, si es positivo o no para el autor que reproduzcamos sus temas, ya que a priori, no van a recibir ningún tipo de monetización. Desde mi punto de vista, es positivo porque se les da visibilidad, pero es cierto que, todas las plataformas deberían tener una opción como la que desde hace muy poco tiempo plantea YouTube, y es que si el video recibe monetización una parte proporcional de dicho video vaya destinada a los autores de los temas. Pero legalmente no debemos preocuparnos, puesto que, hoy en día son las plataformas de retransmisión las que detectan que temas suenan y si infringes sus términos, ellas mismas bloquean tu contenido o silencian el fragmento en cuestión.

Considero que este trabajo no tiene ningún tipo de impacto negativo en cuanto diversidad para ningún colectivo, no se excluye a nadie, sin importar ninguna de sus cualidades, puesto que se trata de un trabajo digital que se va a poder consumir desde todo tipo de dispositivos.

1.4. Enfoque y método seguido

En este proyecto vamos a desarrollar no tan solo un producto nuevo, si no, varios productos nuevos, los cuales van a coexistir entre sí. Es por ello que, pese a ser capaz de realizar la totalidad de las entregas, vamos a priorizar el realizar primero las tareas más sencillas, para que psicológicamente, veamos que el proyecto avanza.

1.5. Planificación del trabajo

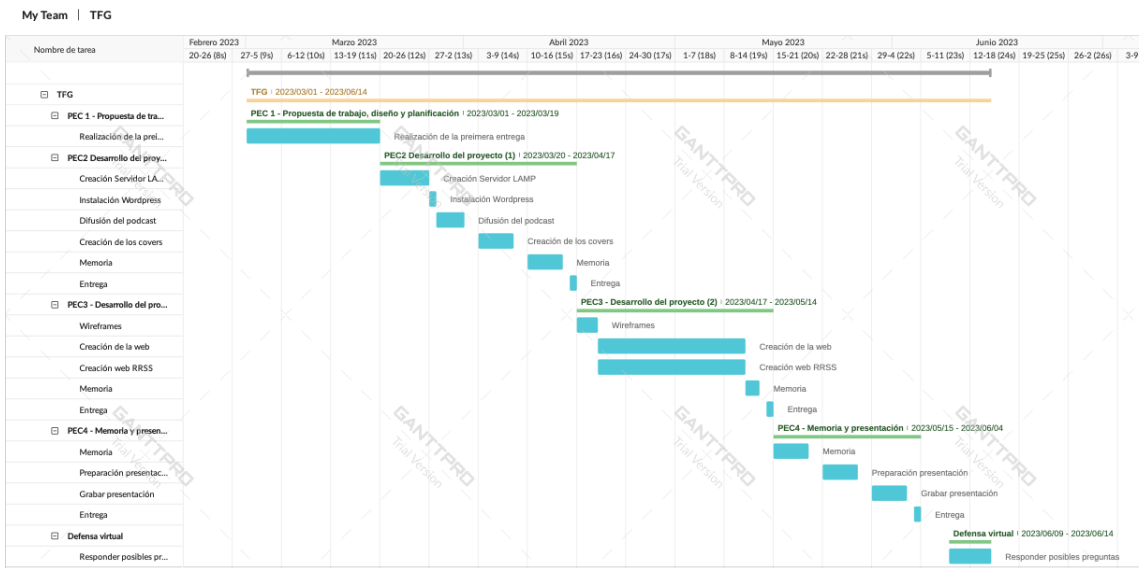


Figura 1: Diagrama de Gantt

1.6. Breve resumen de productos obtenidos

- Servidor LAMP + FTP: Un servidor web totalmente funcional, además de FTP para poder subir contenido de manera rápida y sencilla.
- Web de artista completa hecha en Wordpress: Accesible desde <https://erikatia.sytes.net>
- Web RRSS hecha en HTML y CSS: Su finalidad es la de la difusión en redes sociales cuando se lancen episodios nuevos en el podcast, puesto que esta tendrá un enlace a todas las plataformas donde estará disponible el programa. Esta, estará accesible desde: <https://erikatia.sytes.net/rrss>
- Covers personalizados para cada episodio y para el programa.
- Podcast disponible en las principales plataformas de podcasting.

1.7. Breve descripción de los otros capítulos de la memoria

En esta sección voy a detallar resumidamente el contenido de los siguientes capítulos de mi TFG:

- Materiales y métodos: Capítulo donde detallo de todos los materiales que he requerido para el desarrollo del proyecto, indicando las tecnologías, protocolos, plugins, temas, hardware y software utilizados.
- Resultados: Sección donde detallo los productos finales obtenidos, digamos que es un mero escaparate de los resultados.
- Problemas: Pese a no estar en la plantilla me ha parecido necesaria la inclusión de este apartado, es donde detallo problemas que me han surgido durante el desarrollo del proyecto.
- Conclusiones y trabajos futuros:
- Glosario: Apartado donde explico el significado de palabras técnicas que aparecen a lo largo de la memoria.
- Bibliografía: Capítulo donde detallo webs donde he consultado información cuando con mis conocimientos no han sido suficientes para el correcto desarrollo.
- Anexos: Sección donde indico más al detalle paso por paso como he llegado a obtener los productos finales que presento.

2. Materiales y métodos

2.1. Máquina virtual

Para el desarrollo principal del proyecto, el objetivo es realizar una página web, estando en este punto, tenemos varias opciones, entre las cuales está la simple utilización de un servicio de hosting, hay multitud de opciones como puede ser Hostinger, hostgator, hostalia, etc. Como he comentado hay multitud de ellos, esto simplifica mucho el trabajo, debido a que un servicio de terceros nos facilitaría un servidor con todos los servicios necesarios, que son Apache, servicio de BBDD y PHP, todos estos servicios deben ser funcionales, y plataformas como las anteriormente mencionadas automatizan esta tarea. No es lo que queremos. Por ese motivo necesitamos crear un servidor que nos haga todas estas funcionalidades, para ello tenemos otras dos opciones, que son, tener un PC físico/máquina virtual en local corriendo, o bien, tratar de utilizar una máquina virtual en la nube, que es el caso que hemos escogido.

He seleccionado esta tecnología por un motivo muy sencillo, la alta disponibilidad y la externalización de recursos. Si utilizásemos una máquina local y quisiéramos que el servicio esté levantado 24/7 deberíamos tener el PC siempre encendido, de la manera que propongo externalizamos el recurso del PC, pero mantenemos el hecho de que tenemos el control total del equipo.

Tras investigar las mejores opciones de virtualización disponibles, finalmente me decidí por Amazon AWS, debido a que la empresa que a la cual pertenece este servicio personalmente me transmite mucha confianza, además de tener un plan gratuito de máquina virtual con 750h/mes, lo cual a nosotros ya nos es suficiente.

Los recursos gratuitos del servidor de Amazon durante el primer año son:

- 750h/mes ($31 \text{ días} * 24\text{h} = 744\text{h}$) es decir podemos tenerlo corriendo las 24h del día.
- 30GB de almacenamiento.

Pasado el primer año nos costará el precio que tenga el procesador por hora que necesitemos. En nuestro caso es el t4g.small, cuenta con las siguientes características:

- 2 vCPU
- 2GiB Memoria
- Generación vigente de procesadores.

Su precio es de 0,0188 USD/hora, si tenemos el servidor encendido las 24h nos costaría 14€/mes mantenerlo activo.

Una vez tengo eso, he decidido optar por un Ubuntu Server como sistema operativo anfitrión de la máquina virtual, el motivo principal de esta decisión es que he trabajado en muchas ocasiones con otras versiones del sistema operativo, más concretamente con la versión de escritorio, además abarca un amplio abanico de software compatible con él, incluyendo los paquetes que necesito, que son:

- Apache
- MySQL Server
- PHP
- phpMyAdmin
- Wordpress
- FTP

Por todo esto, considero que es un sistema muy completo con el que poder realizar este proyecto.

He decidido utilizar Wordpress para este proyecto, porque me parece un CMS muy completo que cuenta con una interfaz lo suficientemente “user friendly” como para poder realizar este proyecto, con relativa soltura. Además, cuenta con un catálogo de plugins muy amplio, que abarcará todas las funcionalidades de la página web.

2.2. Album Covers

Los álbum covers es una tarea secundaria de este proyecto, que pese a que no es importante en el objetivo del mismo, si aporta ciertos aspectos que necesito para dar la impresión que busco, que es la de la profesionalidad. Es por ello, que también voy a crear unos álbum covers del podcast general y de todos y cada uno de los episodios mensuales que lanzo.

Para esto vamos a utilizar la herramienta Adobe Illustrator, y una plantilla de la plataforma Freepik como base para los covers. Voy a utilizar una plantilla, porque no es una tarea principal del proyecto y considero que no merece la pena invertir una gran cantidad de tiempo en ella para tratar de conseguir un resultado semejante al que obtenemos utilizando una platilla como base, la plantilla en cuestión que he utilizado es la de la siguiente imagen.



Figura 2: Platilla Freepik

2.3. Difusión Podcast

Como he comentado anteriormente, otro objetivo del proyecto es que el podcast abarque mucho más público, y esto se consigue estando en más plataformas de podcasting entre otras cosas. Esto lo consigo utilizando un fichero RSS.

La primera decisión que tenemos que tomar es donde alojamos el podcast, si en mi servidor web alojado en la máquina virtual, o en iVoox. La decisión final ha sido hacerlo en iVoox y de ahí generar el RSS para la redifusión al resto de plataformas.

La decisión ha sido por un tema de recursos, los recursos de almacenaje en la máquina virtual son limitados a 30GB de uso gratuito, actualmente mi podcast cuenta con algo menos de 100 episodios y cada uno de ellos ocupa unos 150MB su fichero .mp3, es decir, alrededor de 15GB solo en archivos de audio, está claro que el espacio es bastante finito si tenemos en cuenta que es un servicio que quiero que perdure en el tiempo.

2.4. Wordpress

Para la web de artista se han necesitado varios paquetes, además de utilizar el CMS Wordpress como base de todo. Ahora voy a detallar a parte de Wordpress que otros paquetes he necesitado.

En primer lugar, vamos a tratar la plantilla, algo primordial elegir de lo primero dentro del mundo Wordpress, puesto que elegir una buena plantilla condiciona el desarrollo del proyecto.

La plantilla escogida ha sido DIVI, es una plantilla que ya había adquirido y utilizado en un proyecto más básico que desarrollé para el negocio de un conocido. Me pareció una plantilla muy versátil con multitud de opciones que me facilitaban el tipo de web que quería crear, además de contar con un constructor visual que hace la experiencia de creación de las páginas unitarias mucho más llevadera.

En segundo lugar, los plugins adicionales, que en este caso han sido:

- Better Search Replace: Este lo he utilizado para realizar los cambios en la base de datos de los enlaces cuando pasé de utilizar la web entrando directamente por IP publica a usar el nombre de host que finalmente utilicé que es erikatia.sytes.net. El plugin lo que hace es buscar en todas las tablas de la base de datos un texto que le ponemos y lo reemplaza por otro.
- Smash Ballon Instagram Feed: Este plugin lo he utilizado para insertar el feed de mi Instagram en la página inicial y en la página social.
- WP Mail SMTP: Con este lo que he hecho, es enlazar el formulario de contacto de una de las páginas con el SMTP de Google para poder realizar el envío de los correos, si no, al hacer uso del formulario, este no enviaba nada.

Por último, los iframes, que son el código insertado de fuentes externas que utilizo para integrar los reproductores de Apple Podcasts y de iVoox.

Estos son extraídos de las siguientes fuentes:

iVoox cuenta con la opción de generar el reproductor integrado dentro de la propia página del programa > Compartir > Reproductor, una vez ahí ya nos facilita el código. Si seguimos esos pasos en este enlace veremos a lo que me refiero, aun así, adjunto captura.

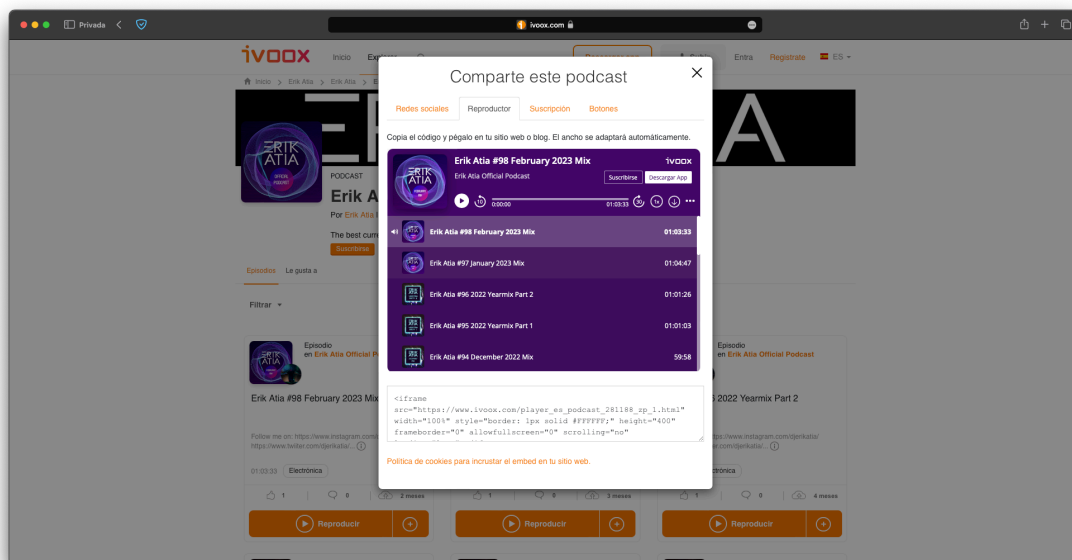


Figura 3: Iframe iVoox

Apple Podcasts: Apple cuenta con una herramienta llamada Apple Marketing Tools, y dentro de la sección de Podcasts, podemos sacar el iframe de nuestro programa. Adjunto captura.

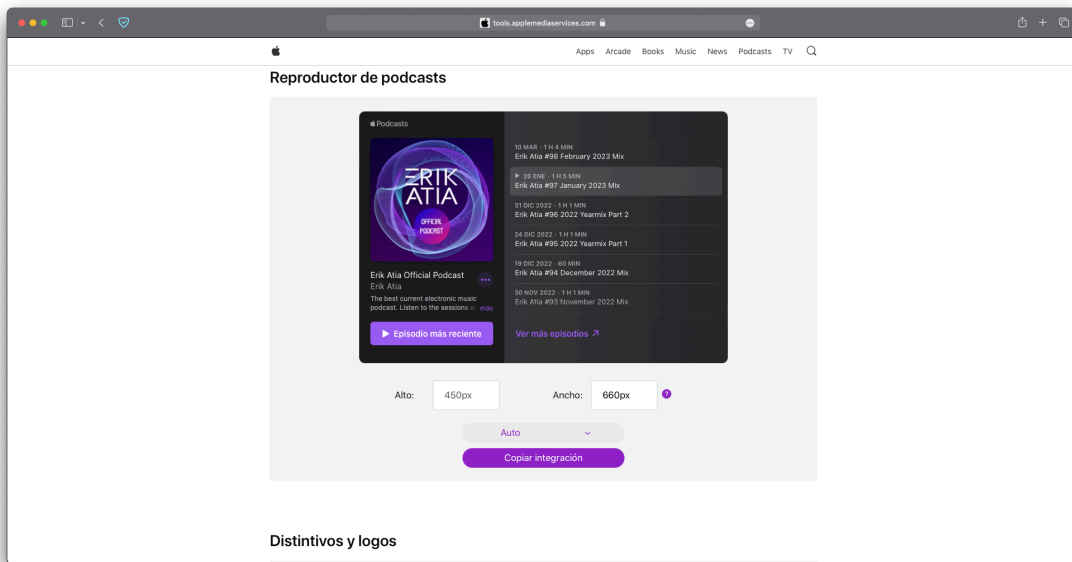


Figura 4: Iframe Apple Podcasts

2.5. Web RSS (HTML + CSS)

Consta de un HTML muy sencillo basado con DIVs, adjunto captura del código.

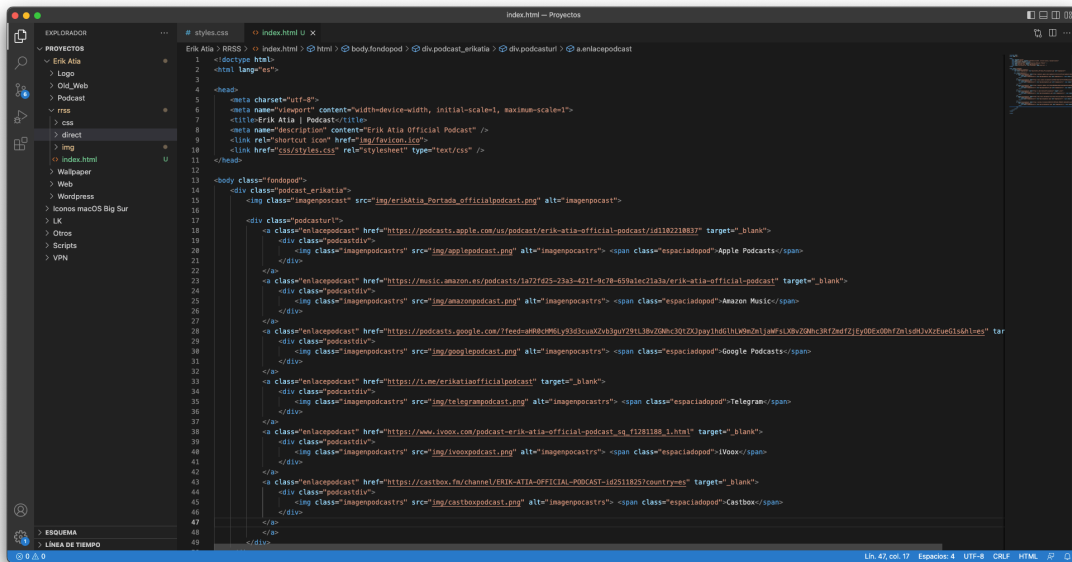


Figura 5: Código HTML

Y de un CSS sencillo, que controla los estilos, controlando también los estilos cuando pasamos el mouse por encima de los enlaces, adjunto también captura.

```

1 /estilos generaliz para 1500w
2 @import url(https://fonts.googleapis.com/css?family=Roboto:400,300,600,400italic);
3
4 {
5   margin: 0;
6   padding: 0;
7   font-family: 'Roboto';
8   line-height: 1.6;
9 }
10
11 /estilos para la pagina de podcast/
12
13 .imagenpodcast {
14   display: block;
15   margin: auto;
16   width: 100px;
17   box-shadow: 2px 2px 10px #888;
18 }
19
20 .podcasturl {
21   text-align: left;
22   width: 40%;
23   margin: auto;
24 }
25
26 .podcasturla {
27   text-decoration: none;
28   color: #333;
29 }
30
31 .podcasturlhover {
32   color: #000;
33   text-decoration: none;
34 }
35
36 .podcastdiv {
37   background: #fff;
38   display: flex;
39   align-items: center;
40   border-radius: 10px;
41   margin-bottom: 10px;
42   box-shadow: 2px 2px 10px #888;
43   margin-top: 10px;
44   transition: all .5s;
45 }
46
47 .podcastdivhover {
48   background: #222;
49   transition: all .5s;

```

Figura 6: Código CSS

2.6. NOIP

NOIP es una plataforma que nos ofrece la posibilidad de crear un hostname y enlazarlo a la IP pública de nuestro servidor, de manera gratuita, siempre y cuando renovemos el hostname cada mes, si no queremos esta limitación sí que será necesario pagar su servicio. Adjunto captura del panel de control, con el dominio creado y como apunta a nuestra IP Pública.

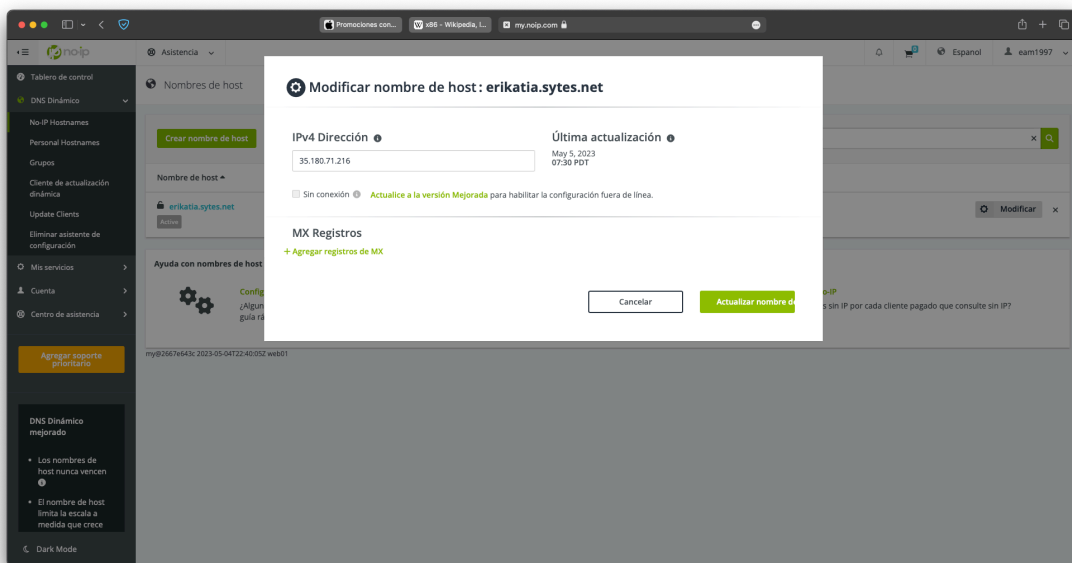


Figura 7: NOIP

2.7. SSL For Free

Tras conseguir hacer funcionar el servidor, y la página web mediante el hostname anteriormente mencionado, me encontré en la situación de que obviamente solo era visible mediante http y no mediante https, esto algo que nos aporta seguridad y profesionalidad, puesto que la conexión con el servidor se vuelve cifrada y no se transmite mediante texto plano. Si generaba el certificado SSL necesario para habilitarlo yo mismo, teníamos el problema de que el navegador lanzaba un mensaje indicando que dicho certificado no había sido aprobado por ninguna CA, así que decidí generarlo con esta plataforma, puesto que me permite hacerlo de manera gratuita siempre y cuando lo renueve manualmente cada 90 días. Como veremos en la siguiente captura, consta el actual certificado, para su dominio y su próxima fecha de caducidad.

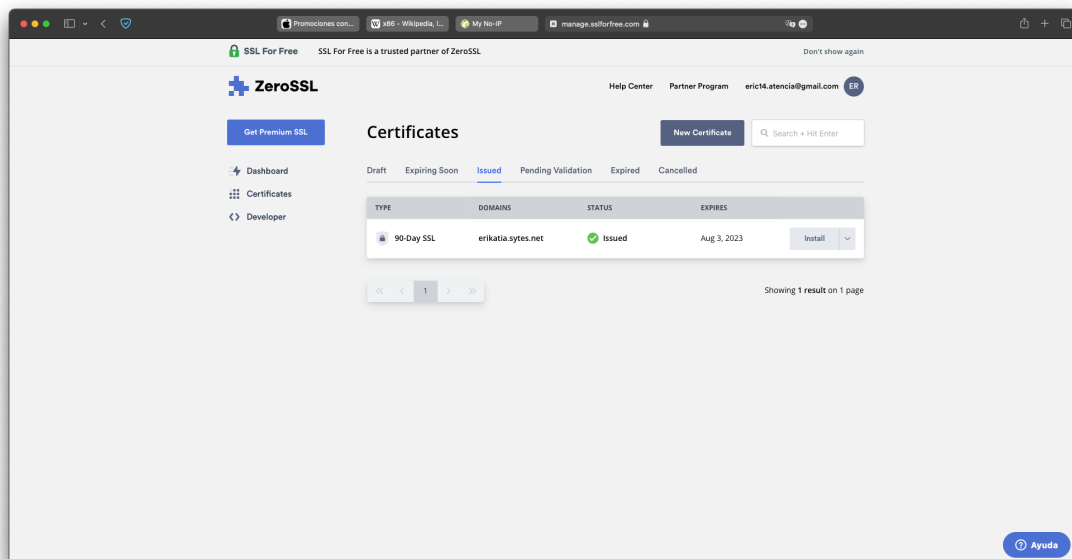


Figura 8: SSL For Free

3. Resultados

3.1. Servidor LAMP + Wordpress

Gracias a la metodología seguida en el apartado anterior ahora contamos con una máquina virtual en Amazon AWS, con un Ubuntu Server 22.04 LTS corriendo los siguientes servicios:

- Apache
- MySQL Server
- PHP
- phpMyAdmin
- Wordpress
- FTP

Siendo capaz de acceder desde cualquier parte, ya que se ha permitido el tráfico web tanto en el firewall interno de la máquina como en el que nos deja gestionar en el panel de administración de Amazon AWS.

3.2. Album Covers

Tras coger los elementos que he considerado útiles de la plantilla comentada anteriormente, y combinarlos con mi logo, he llegado a un producto final que tiene el siguiente aspecto.

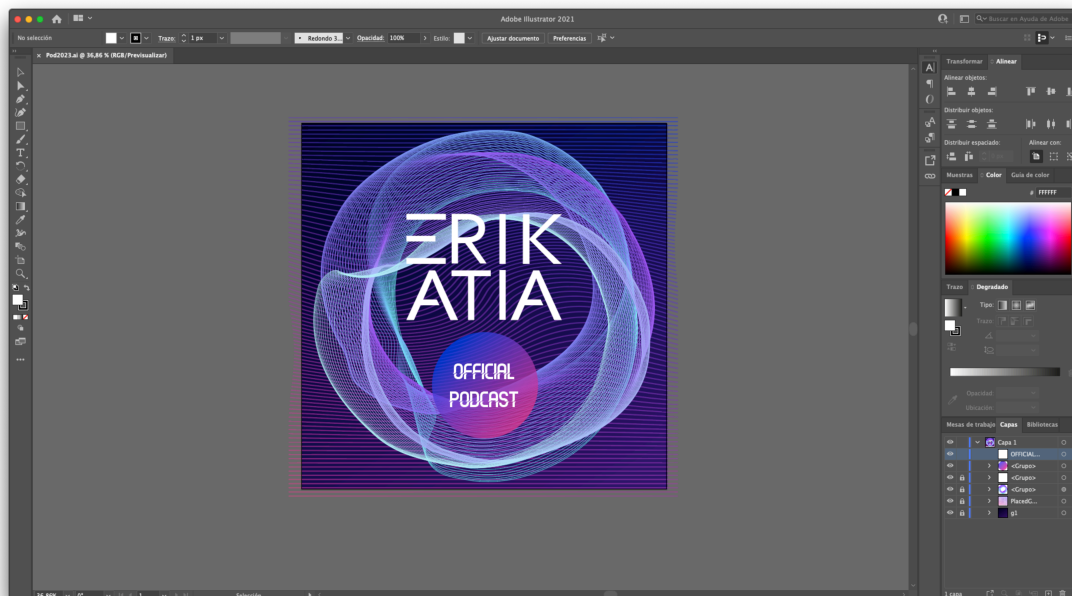


Figura 9: Portada Podcast

Una vez conseguido esto, he creado una portada para cada episodio de cada mes, como vemos en la siguiente imagen.

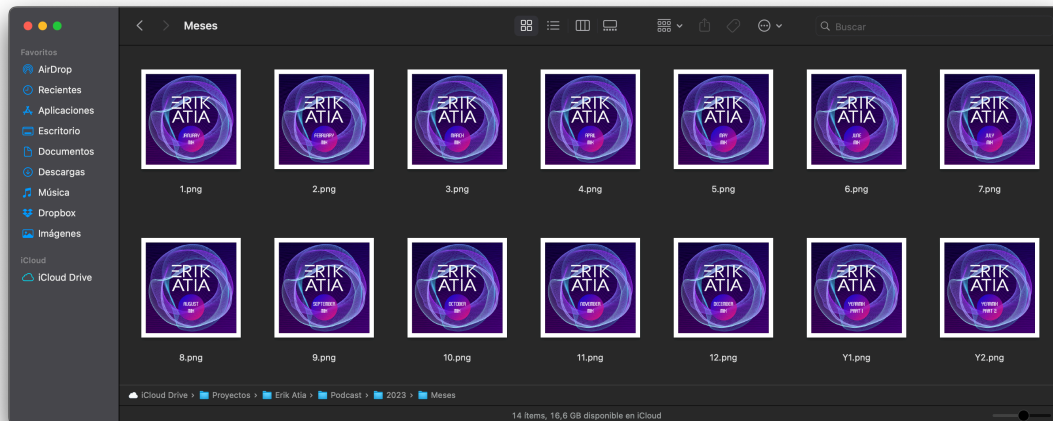


Figura 10: Episodios Podcast

3.3. Difusión Podcast

Tras haber difundido el enlace RSS de iVoox en las principales plataformas, he conseguido que el podcast esté disponible y se actualice automáticamente en los siguientes servicios:

- Apple Podcast

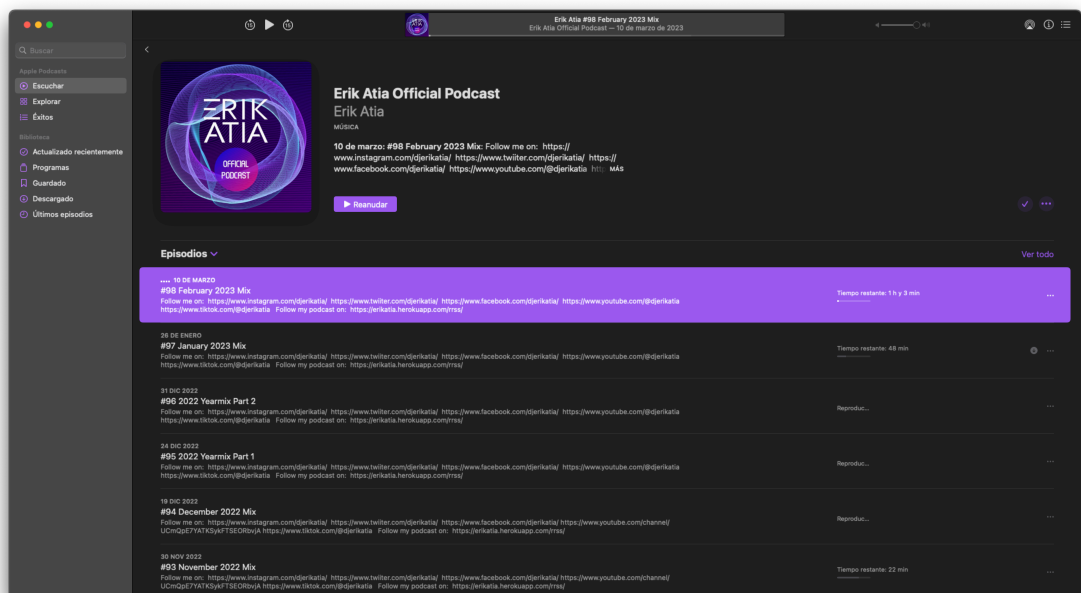


Figura 11: Apple Podcasts

- Amazon Music

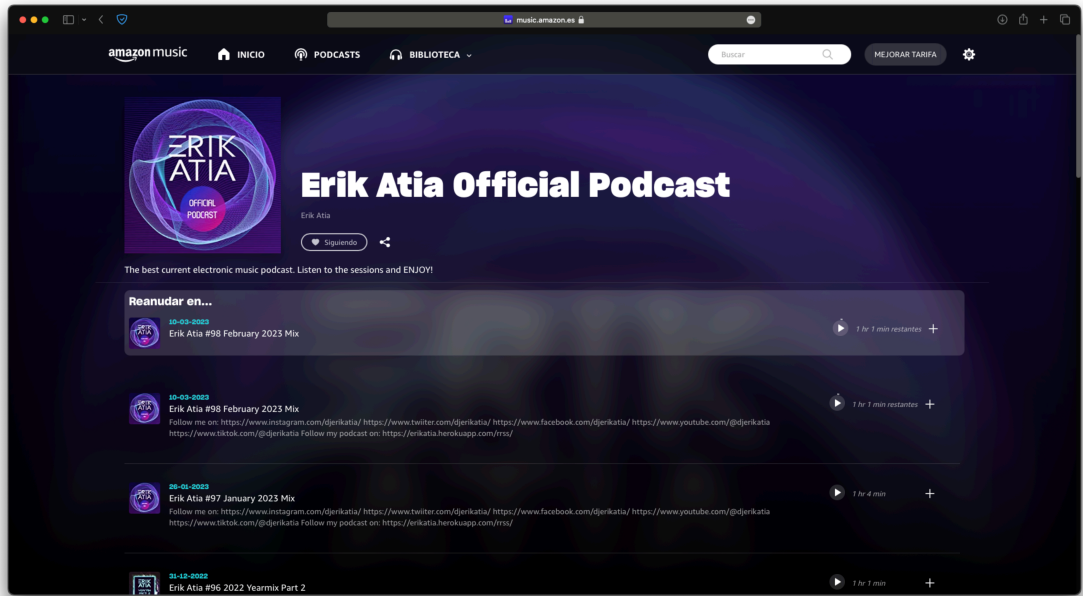


Figura 12: Amazon Music

- Google Podcasts

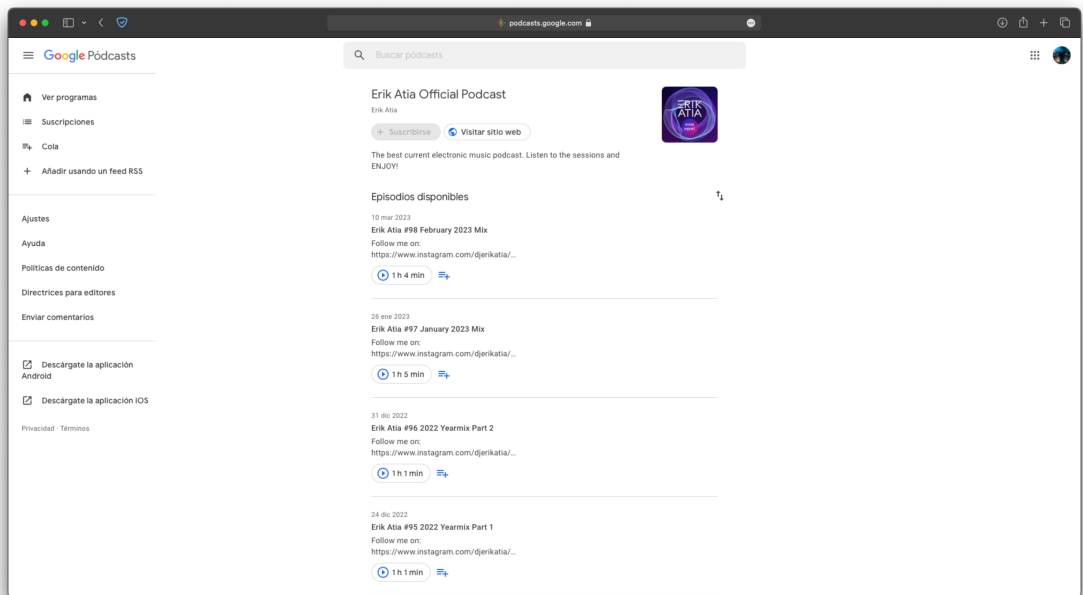


Figura 13: Google Podcasts

- Castbox:

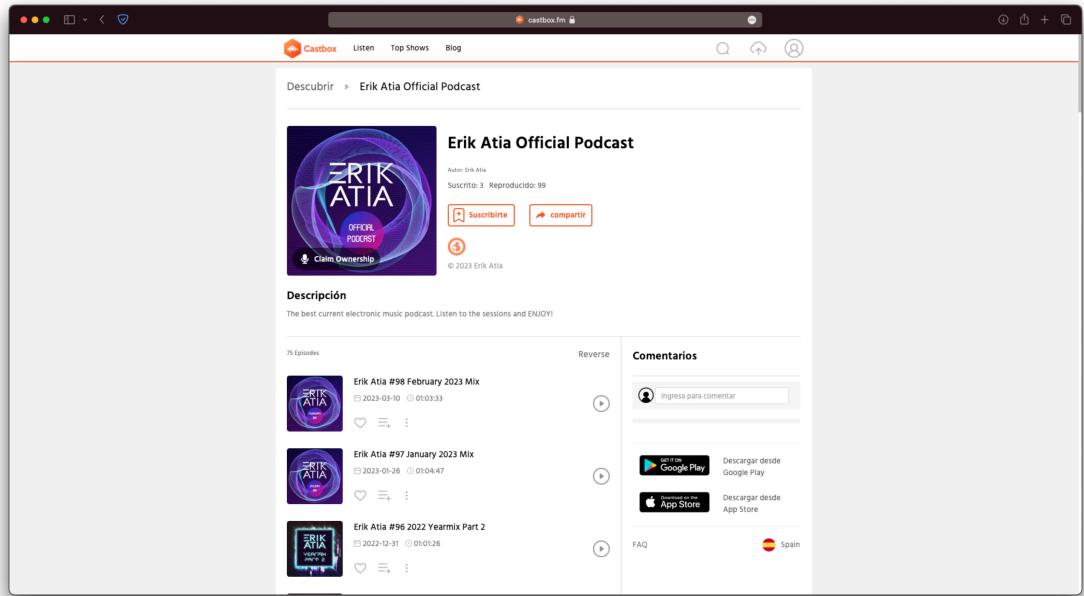


Figura 14: Castbox

3.4. Web artista (Wordpress)

Web de artista completa desarrollada en Wordpress (<https://erikatia.sytes.net>), paso a explicar más detalladamente el contenido de las páginas y dejo por aquí algunas capturas de pantalla de sus páginas.

Home:

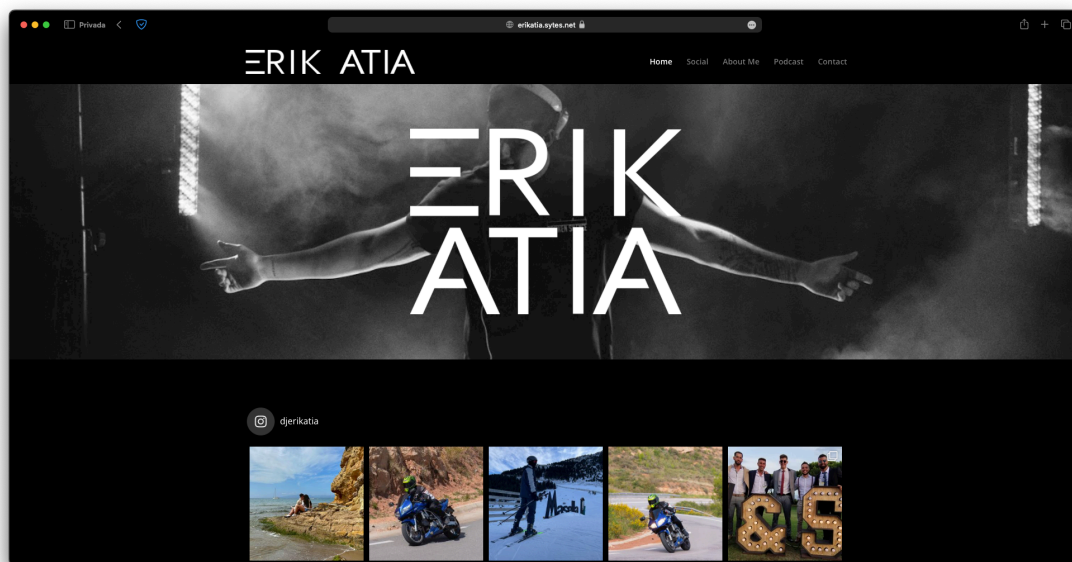


Figura 15: Home

Contamos con un home que su función principal es “dar la bienvenida”, les presentamos la marca “Erik Atia”, ofreciendo el logo y una imagen de fondo, era importante que congeniase bien con la paleta de colores que quería ofrecer, es decir unas tonalidades entre negro y grises. Posteriormente ofrezco una ligera vista al feed de Instagram para que se hagan a la idea de que tipo de persona podemos ser y más abajo (no presente en la imagen) una ligera vista al podcast para que vean sin necesidad de cambiar de página que tipo de contenido ofrezco.

Social:

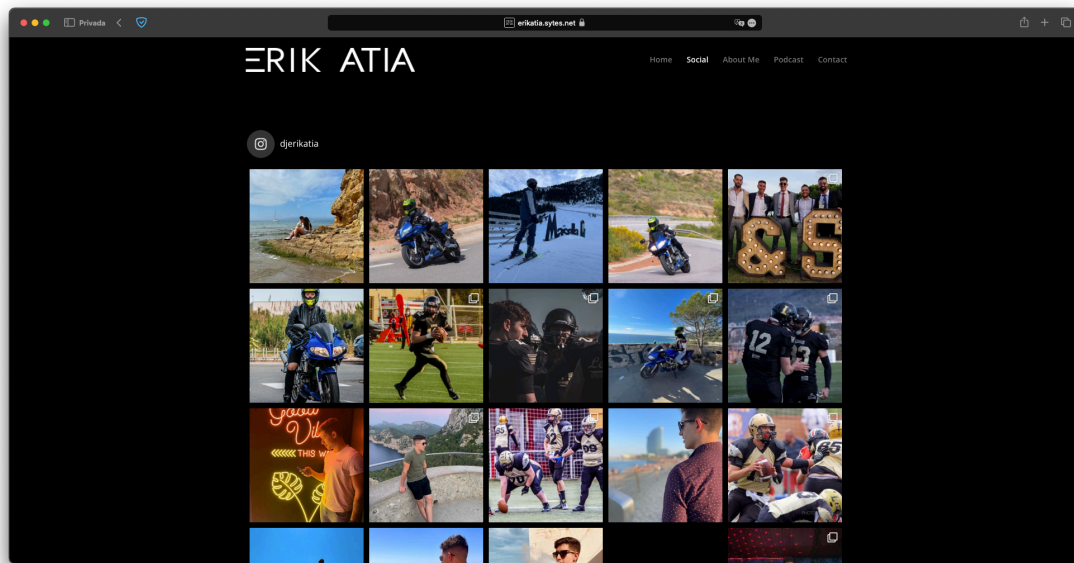


Figura 16: Social

Esta página simplemente es con dos finalidades, la primera es extender un poco más la primera imagen que hemos dado en el Home, dando aquí más información sobre el perfil de persona que somos, dejando ver más publicaciones, y el segundo motivo es una clara intención de publicitar nuestro perfil para tratar de conseguir algún seguidor.

About Me:

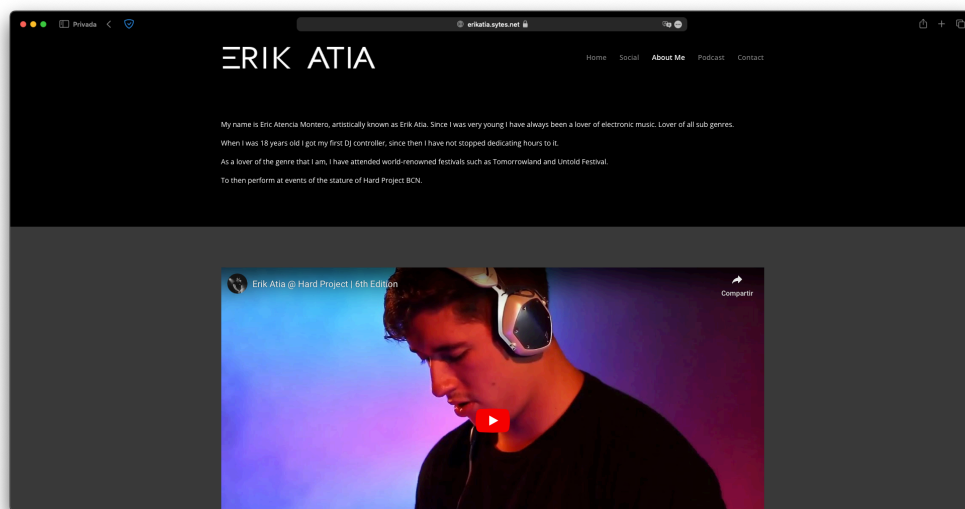


Figura 17: About Me

En esta página explico un poco de mi historia, cual es mi trayectoria y como he llegado hasta aquí, además de ofrecer un video incrustado de una de mis sesiones.

Podcast:

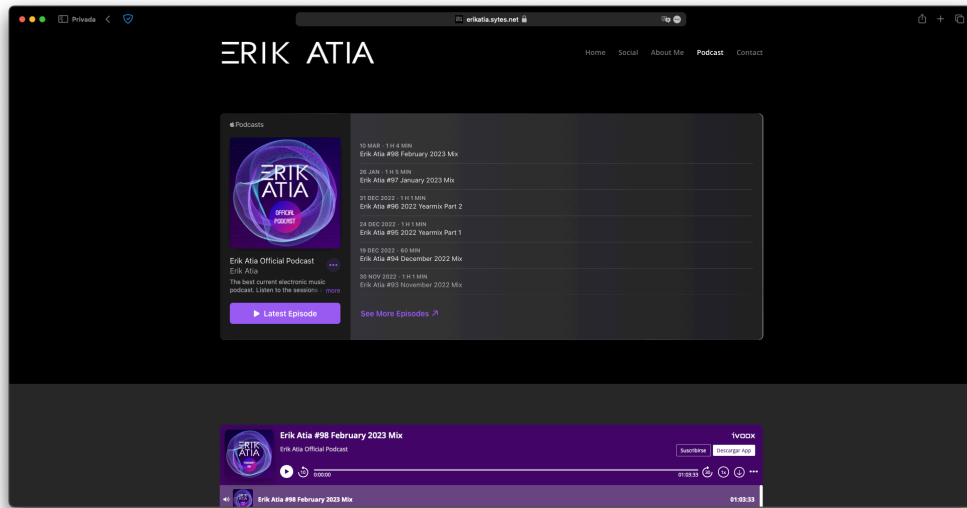


Figura 18: Podcast

Aquí es donde hacemos énfasis absoluto al podcast, teniendo incrustados los reproductores de Apple Podcast e iVoox, aquí sí que busco conseguir oyentes hacia mi contenido.

Contact:

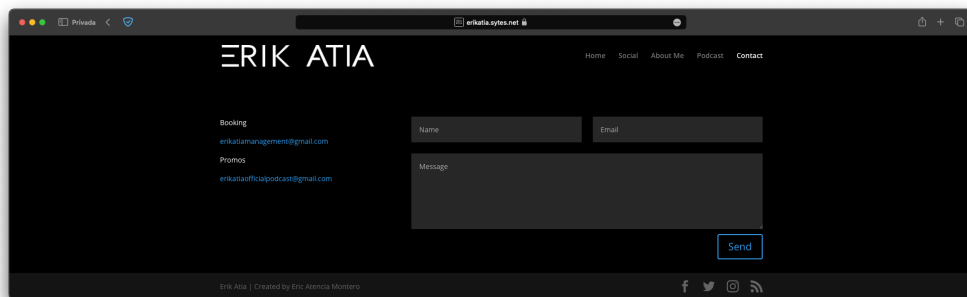


Figura 19: Contact

Y en esta página de contacto es simplemente porque considero totalmente necesario que los usuarios tengan una manera rápida y sencilla de ponerse en contacto conmigo.

3.5. Web RRSS (HTML + CSS)

Web desarrollada para ser vista con dispositivos móviles puesto que es con los que consumimos contenido en redes sociales. (<https://erikatia.sytes.net/rrss>)

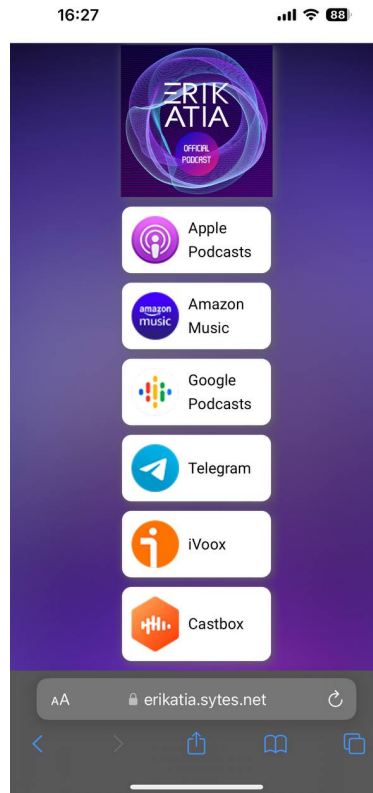


Figura 20: RRSS

Ha sido desarrollada en HTML + CSS, su función es algo parecido a lo que hace linktr.ee, es homogenizar todos los enlaces a las plataformas de manera sencilla, de manera que cuando hagamos publicidad del podcast, con un único enlace cubramos todas las plataformas, siendo en el usuario el que elige en este paso.

Problemas

No ha sido un proyecto que me haya dado muchos problemas graves en rasgos generales. Han sido en mayor medida problemas minoritarios en cuanto a configuraciones varias en el servidor Linux sobre todo.

Pero, es cierto que me he encontrado con un problema que sí que me ha costado solventar. Ha sido el de los recursos del servidor, me explico. Una vez instalado Wordpress al final de la primera entrega parcial del proyecto, parecía que iba todo genial, pero el problema vino cuando comencé a trabajar sobre Wordpress y comencé a solicitar carga de trabajo al servidor. El problema fue que mientras desarrollaba la web se iba quedando congelado el servidor, y solo podía volver a trabajar reiniciándolo, la web corría sin problema en modo usuario, pero cuando entraba a la administración del servidor se bloqueaba a la que llevaba 2 minutos trabajando

En ese punto comencé a probar cosas para solventar la incidencia, intenté reinstalar todo, pensando que quizá me había equivocado en algún punto, en vista de que esto no funcionaba, pensé en instalar MAMP en mi ordenador local, desarrollar la web entera y posteriormente migrarla al servidor en la nube, pese a que no me parecía práctico para luego mantener la web era una idea que tenía para salvar el paso. Esta opción me dio problemas luego con los enlaces internos, y no era posible navegar dentro de la web, estuvo unas horas intentando solventarlo, pero no lo logré y como no era una solución práctica, finalmente me decanté por volver a intentar solventar el tema del servidor sin tener que pagar por unos recursos fuera de la capa gratuita de Amazon AWS.

Es ahí cuando decidí eliminar la máquina virtual entera y rehacerla con otra arquitectura. En un principio la máquina virtual contaba con un procesador con arquitectura x86, que es la arquitectura más común para este tipo de cosas, pero investigando vi que me daba la opción de realizarlo con un único procesador concreto de arquitectura ARM, es decir, la arquitectura de RaspBerry Pi, por ejemplo, o sin ir más lejos, los procesadores Apple Silicon que sacaron hace 1 año aproximadamente. Este tipo de procesadores tienen la cualidad de tener un muy bajo consumo de recursos, esto nos da la capacidad de poder rendir más con menos recursos. Tras crear la máquina con estos recursos y posteriormente instalar todo de nuevo conseguí que todo funcionase sin tener ningún bloqueo.

Conclusiones y trabajos futuros

Tras haber finalizado el transcurso del proyecto, he obtenido todos los productos finales que deseaba, falta por ver si tras haber dado más visibilidad y profesionalidad al proyecto Erik Atia, consigo tener más viralidad y repercusión en el número de actuaciones que realizo como artista.

Siendo críticos con todos los resultados obtenidos, creo que he logrado cumplir con todos los objetivos, cierto es que ciñéndome al marco de las capas gratuitas, no es del todo cómodo estar renovando ciertos aspectos del proyecto cada cierto tiempo, es mucho más fácil y sencillo para la finalidad que buscaba contratar un servicio hosting, delegando así el mantenimiento del servidor y la gestión del certificado SSL, pero me parecía interesante para este proyecto llevar a cabo de manera propia todos los servicios.

Creo que la planificación si ha sido algo que he conseguido cuadrar muy acertadamente, puesto que he dejado el espacio de tiempo necesario para el correcto desarrollo de todos los puntos del proyecto, dejando incluso cierto tiempo para posibles problemas que pudieran surgir en el desarrollo del mismo. Así que teniendo en cuenta mis capacidades y conocimientos, creo que la planificación del proyecto has sido muy correcta.

En cuanto a impactos de sostenibilidad, no podemos saber con exactitud la contaminación que produce nuestro servidor, no es un dato que nos facilite Amazon AWS, pero podemos tomar como referencia los datos aportados en el apartado 1.3 del proyecto.

En cuanto a temas éticos tratando así las licencias de uso, lo que hemos hecho es delegar a las plataformas donde subimos el contenido la gestión de los derechos de la música, ya que son ellas las que pagan por reproducir contenido con derechos dentro de sus plataformas. Hemos excluido las plataformas que no permiten este tipo de reproducción como son Spotify y YouTube.

Los objetivos del trabajo considero que se han cumplido todos, pero es cierto que es un proyecto escalable. A futuro creo que migraré la página web a un servicio de hosting, como he comentado anteriormente, no me parece práctica la gestión de tantos procesos por separado, renovando incluso algunos cada cierto tiempo, esta filosofía de autogestionarme todos los servicios, supondría que debería pagar el coste del servidor en la nube pasado el año, el certificado SSL y el hosting, todo esto sumado implica un presupuesto mucho mayor que si nos ceñimos a delegar el servicio a un servicio de hosting como puede ser Hostinger, en este caso solo debería gestionarme la web en sí misma.

En resumen, me ha parecido un proyecto muy satisfactorio con el que he aprendido mucho de la administración de un servidor de este tipo, y sobre la creación de un contenido multimedia de estas características.

4. Glosario

- Apache: es un software de servicio web, es de código abierto y es utilizado páginas web en alrededor del 46% de sitios web del mundo.
- MySQL Server: es un sistema de bases de datos relacionales, es de código abierto y nos permite crear, almacenar y editar información de nuestras bases de datos.
- PHP: es un lenguaje de programación de código abierto, principalmente se utiliza en para aplicaciones web.
- phpMyAdmin: es una herramienta de código abierto que nos permite administrar de manera grafica las bases de datos de nuestro MySQL Server
- VM (Virtual machine/máquina virtual): una máquina virtual, digamos que es un software que va a simular que tenemos otro ordenador dentro del nuestro, simulando un hardware concreto y un sistema operativo.
- Amazon AWS (Amazon Web Services): es una plataforma oficial de Amazon, que proporciona servicios en la nube.
- RSS: es un formato de archivo XML utilizado para distribuir contenido actualizado en la web.
- Arquitectura x86: arquitectura de procesador más común hoy en día, presente en prácticamente todos los ordenadores domésticos.
- Arquitectura ARM: arquitectura presente en mini ordenadores, como RaspBerry u ordenadores de la nueva generación de procesadores de Apple.
- FTP: File Transfer Protocol, es un protocolo común de acceso al servidor para la transferencia de ficheros entre cliente servidor, frecuentemente se utiliza en servidores web.
- HTTP y HTTPS: son los protocolos que se utilizan para el tráfico web, la diferencia entre ellos es que http vas sin cifrar y se envía y recibe el tráfico por texto plano y https lo cifra para que sea ilegible ante cualquier intrusión.
- Plugins: Digamos que es una utilidad extra que instalamos a nuestro Wordpress.
- Temas: Es una plantilla ya pre hecha con la estructura visual de la web.

5. Bibliografía

- ComoInstalar.* (s.f.). Obtenido de ComoInstalar: <https://comoinstalar.me/como-instalar-wordpress-en-ubuntu-22-04-lts/>
- DigitalOcean.* (s.f.). Obtenido de DigitalOcean: <https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-20-04-es>
- DigitalOcean.* (s.f.). Obtenido de DigitalOcean: <https://www.digitalocean.com/community/tutorials/how-to-create-a-new-sudo-enabled-user-on-ubuntu-18-04-quickstart-es>
- EnMiMaquinaFunciona.* (s.f.). Obtenido de EnMiMaquinaFunciona: <https://www.enmimaquinafunciona.com/pregunta/23201/como-puedo-copiar-el-contenido-de-una-carpeta-a-otra-carpeta-en-un-directorio-diferente-utilizando-el-terminal>
- Freepik.* (s.f.). Obtenido de Freepik: https://www.freepik.es/vector-gratis/folleto-club-linea-forma-degradado-dinamico-conjunto-portada-discoteca-ondulada-folleto-club-neon-musica-electro-dance-fiesta-trance-dj-festival-sonido-electronico-cartel-evento-techno_26388337.htm#query=dj%20cover&pos
- Gaeapeople.* (s.f.). Obtenido de Gaeapeople: <https://gaeapeople.com/canal-digital/que-es-un-green-hosting-o-alojamiento-web-sostenible/>
- Geekflare.* (s.f.). Obtenido de Geekflare: <https://geekflare.com/es/free-ssl-tls-certificate/>
- Google.* (s.f.). Obtenido de Google: <https://www.google.com/search?client=safari&rls=en&q=sacar+rss+de+iv+vox&ie=UTF-8&oe=UTF-8>
- Hostinger.* (s.f.). Obtenido de Hostinger: <https://www.hostinger.es/tutoriales/renombrar-archivos-linux>
- Hostinger.* (s.f.). Obtenido de Hostinger: <https://www.hostinger.es/tutoriales/que-es-apache/>
- HowToForge.* (s.f.). Obtenido de HowToForge: <https://howtoforge.es/como-instalar-phpmyadmin-en-ubuntu-22-04/>
- Linktree.* (s.f.). Obtenido de Linktree: <https://linktr.ee>

Youtube.

(s.f.).

Obtenido

de

Youtube:

<https://www.youtube.com/watch?v=UNeTOHD21Vc>

6. Anexos

6.1. Creación de la máquina virtual

Una vez nos hemos registrado en la plataforma y hemos entrado con nuestra cuenta, nos deberemos ir al enlace que indica “lanzar una instancia”.

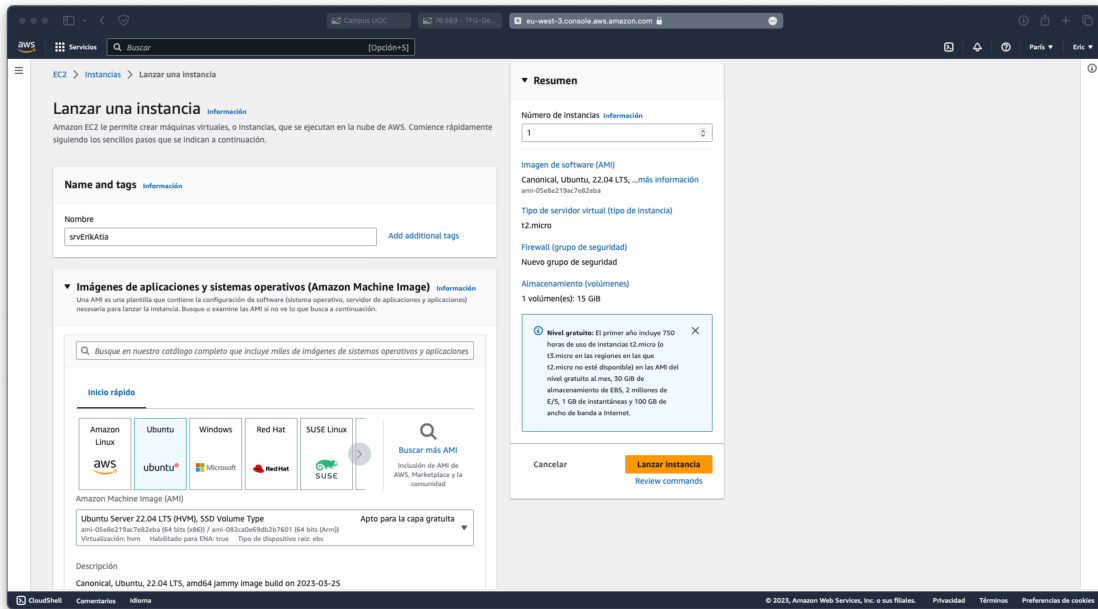


Figura 21: Creación máquina virtual 1

Rellenamos los campos, con los datos necesarios, yo he utilizado srvErikAtia, porque es la instancia que dedicaré a mi web como artista y ese es mi nombre artístico.

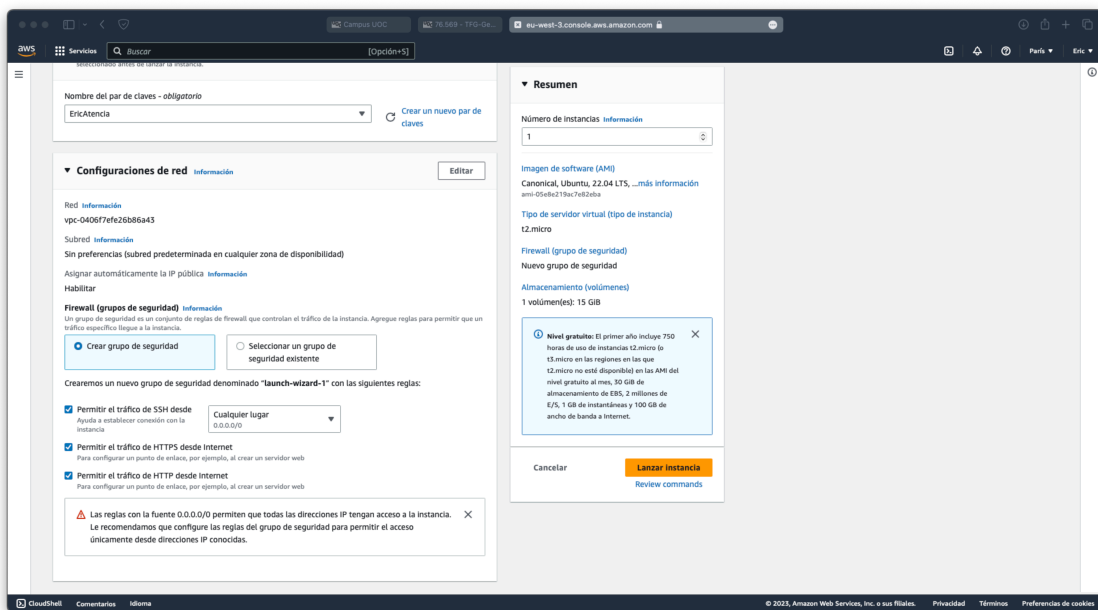


Figura 22: Creación máquina virtual 2

Creamos un nuevo par de claves, indicando el nombre que queremos ponerle, en mi caso EricAtencia, además permitimos el tráfico SSH para poder conectarnos al servidor y permitimos tanto el tráfico HTTP como el HTTPS, para poder hacer uso de los servicios web.

Al generar el fichero de claves, nos dará como resultado lo siguiente:

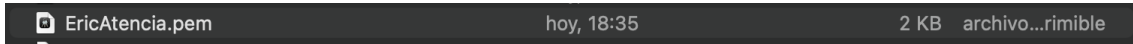


Figura 23: Archivo .pem

Ahora lanzamos la instancia y nos lanzará esta pantalla:

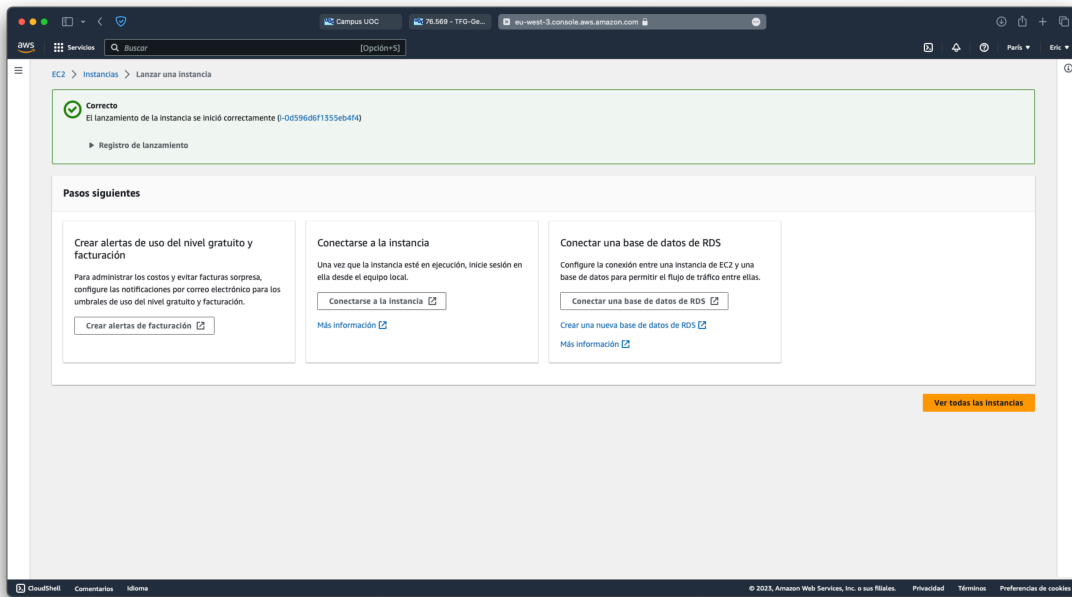


Figura 24: Máquina creada

Y en nuestro resumen de máquinas veremos cómo está en ejecución y algunos detalles de la máquina.

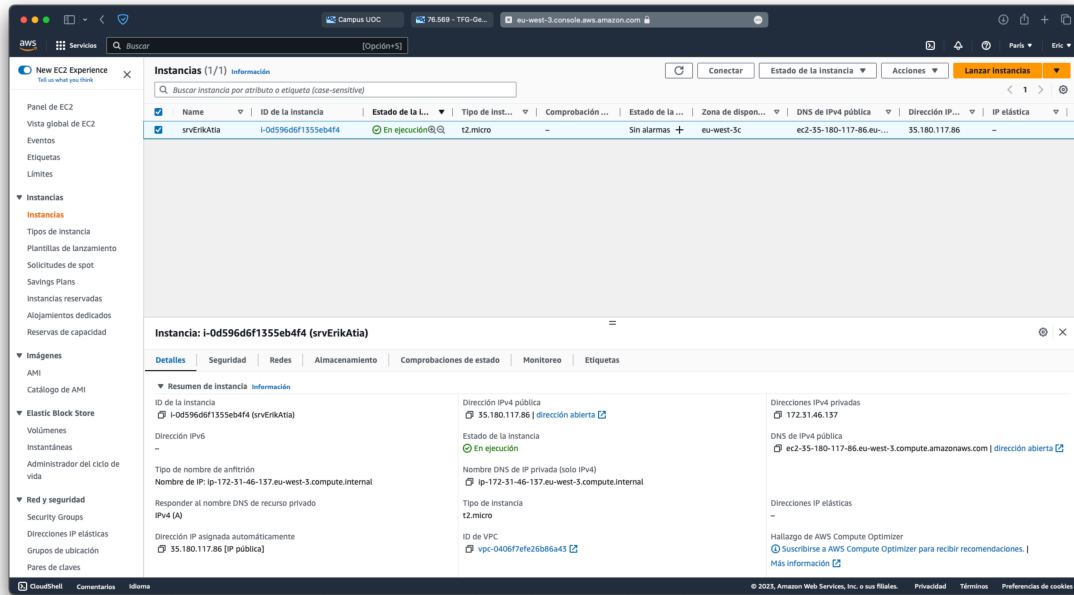


Figura 25: Detalles máquina

6.2. Conexión con el servidor

Bien, ahora nos vamos a conectar por SSH a la máquina virtual, esto lo hacemos vía el siguiente comando en el Terminal.

```
eric@MacBook-Pro-de-Eric:~$ sudo ssh ubuntu@35.180.117.86 -i /Users/eric/Library/Mobile\ Documents/com~apple~CloudDocs/Grado\ Multimedia/5t/2n\ Semestre/TF6/EricAtencia.pem
```

Figura 26: Conexión SSH a la VM

6.3. Actualización del servidor

Una vez estamos conectados a la máquina vamos a proceder a actualizarla. Primero vamos a actualizar los repositorios de paquetes.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get update
```

Figura 27: Actualización repositorios

Y ahora vamos a actualizar cualquier paquete que tenga disponible una versión más reciente.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get upgrade
```

Figura 28: Actualización de paquetes

6.4. Instalación y configuración LAMP

Lo primero va a ser instalar Apache, que es el encargado de correr la página web.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get install apache2
```

Figura 29: Instalación Apache

Ahora vamos a tener que permitir el tráfico Apache en el firewall de nuestro Linux, esto lo haremos con el siguiente comando.

```
ubuntu@ip-172-31-46-137:~$ sudo ufw allow in "Apache"
Rules updated
Rules updated (v6)
```

Figura 30: Firewall Apache

Esto es opcional, pero vamos a instalar el paquete que nos permite ver la IP del servidor y varios datos más de redes que considero útiles.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get install net-tools
```

Figura 31: Instalación paquete redes

Una vez hecho todo esto, si accedemos a la IP pública del servidor mediante el navegador web veremos lo siguiente:



Figura 32: Apache funcionando

Ahora vamos a instalar el servidor MySQL para poder alojar bases de datos, esto será necesario para poder utilizar Wordpress posteriormente.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get install mysql-server
```

Figura 33: Instalación MySQL Server

Lo siguiente será instalar PHP, lo cual también va a ser necesario para el correcto funcionamiento de nuestro Wordpress.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get install php libapache2-mod-php php-mysql
```

Figura 34: Instalación PHP

Por último, vamos a instalar phpMyAdmin para poder gestionar las bases de datos de nuestro servidor de una manera mucho más sencilla, ya que contaremos con una interfaz visual vía web.

```
ubuntu@ip-172-31-46-137:~$ sudo apt-get install phpmyadmin
```

Figura 35: Instalación phpMyAdmin

Accederemos introduciendo en navegador la IP de nuestro servidor /phpmyadmin, y veremos algo así.

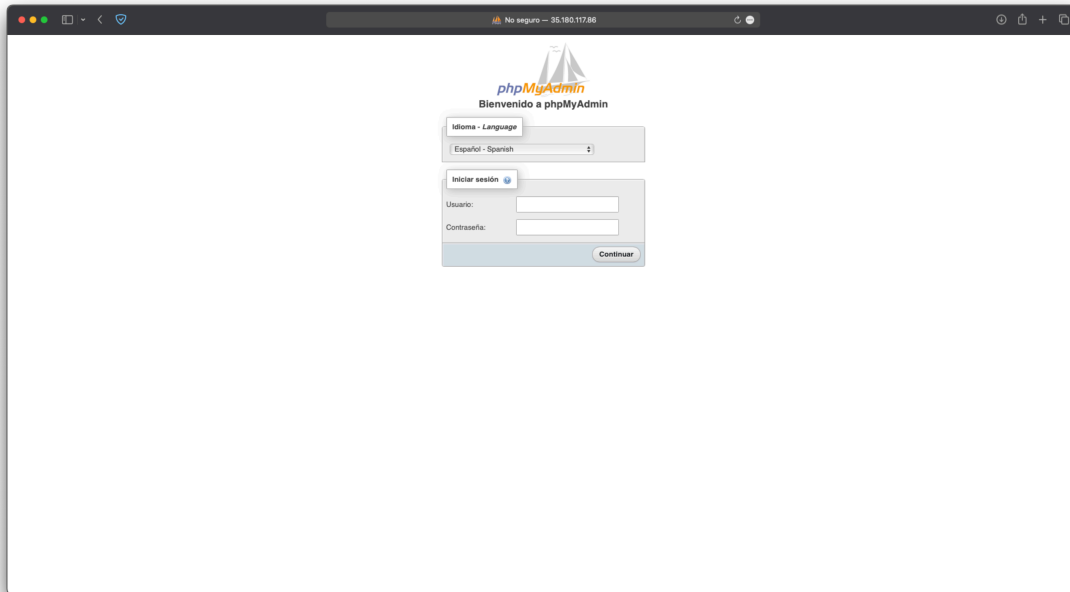


Figura 36: phpMyAdmin funcionando

Bien, ahora necesitamos crear un usuario que pueda acceder a phpMyAdmin, es decir un usuario en MySQL Server, esto se hace accediendo a la consola de este servicio con el comando “sudo mysql”, entonces creamos el usuario de la siguiente forma:

```
[mysql> CREATE USER 'eam1997'@'localhost' IDENTIFIED BY
Query OK, 0 rows affected (0.01 sec)
```

Figura 37: Creación usuario MySQL Server

Tras las palabras “IDENTIFIED BY” deberemos poner la contraseña deseada, esta deberá ir entre comillas simples.

Bien, ahora deberemos otorgarle permisos root a este usuario para que podamos usarlo como usuario gestor.

```
[mysql> GRANT ALL PRIVILEGES ON * . * TO 'eam1997'@'localhost';
Query OK, 0 rows affected (0.02 sec)
```

Figura 38: Permisos MySQL Server

Y aplicamos los permisos:

```
[mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)
```

Figura 39: Aplicar permisos

Ahora, tras iniciar sesión en la interfaz web veremos la pantalla inicial de phpMyAdmin.

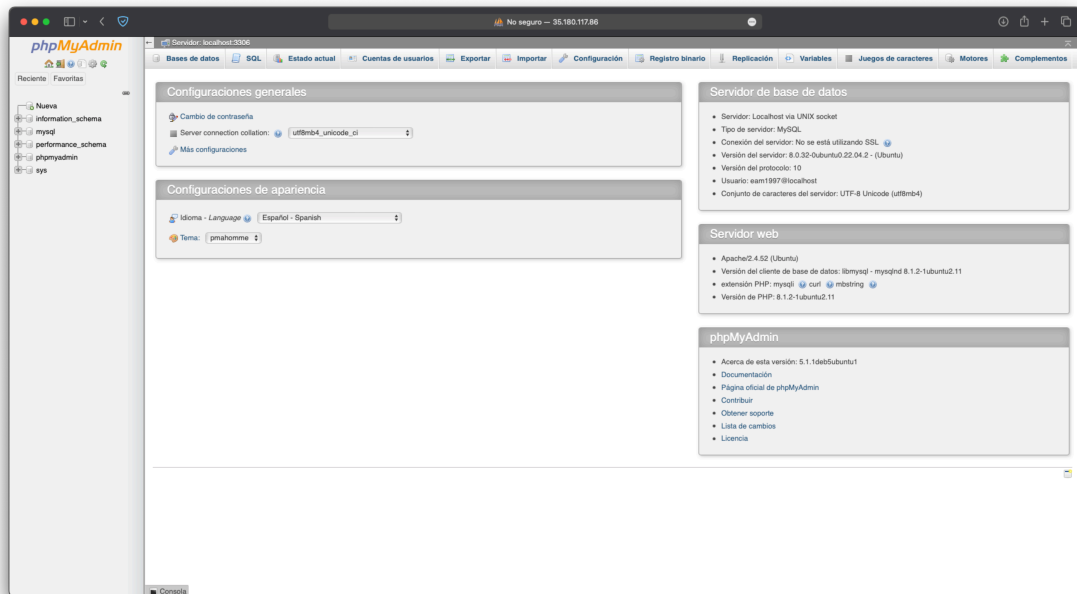


Figura 40: Inicio de phpMyAdmin

6.5. Instalación Wordpress

Lo primero para iniciar este proceso va a ser crear la base de datos para la página web, esto se hará en phpMyAdmin > Bases de datos, allí creamos la base de datos como vemos en la imagen.



Figura 41: Creación BBDD

Ahora descargaremos Wordpress, en mi caso lo he descargado en una carpeta llamada “Eric” que he creado en la raíz del perfil, la descarga se hace de la siguiente forma.

```

ubuntu@ip-172-31-46-137:~/Eric$ wget https://es.wordpress.org/latest-es_ES.tar.gz
--2023-04-10 17:17:36-- https://es.wordpress.org/latest-es_ES.tar.gz
Resolving es.wordpress.org (es.wordpress.org)... 198.143.164.252
Connecting to es.wordpress.org (es.wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 23804314 (23M) [application/octet-stream]
Saving to: 'latest-es_ES.tar.gz'

latest-es_ES.tar.gz  100%[=====] 22.70M  10.4MB/s  in 2.2s
2023-04-10 17:17:39 (10.4 MB/s) - 'latest-es_ES.tar.gz' saved [23804314/23804314]

ubuntu@ip-172-31-46-137:~/Eric$
  
```

Figura 42: Descarga de Wordpress

Para ahorrarnos problemas, vamos a renombrar el index.html por defecto de Apache, de manera que no nos interfiera en los ficheros de Wordpress, yo lo renombro para tenerlo en caso de que nos sea necesario, esto lo hacemos con el simple comando “mv /var/www/html/index.html /var/www/html/index.html.bak”. Ahora, descomprimos el contenido del fichero comprimido que habíamos descargado con el comando “sudo tar xf latest-es_ES.tar.gz”. Para finalizar con el movimiento de carpetas movemos el contenido de la carpeta Wordpress que se nos genera a la ruta “/var/www/html/” esto lo hacemos con “sudo cp -a wordpress/. /var/www/html/”. Y para finalizar con la edición de permisos, deberemos asignar un nuevo propietario a la ruta por defecto de Apache, ya que ahora se deberán poder hacer cambios desde el navegador, esto lo hacemos con “sudo chown -R www-data: /var/www/html”.

Ahora si volvemos a acceder a nuestra IP publica a través del navegador veremos que ya tenemos la pantalla de bienvenida de Wordpress.



Figura 43: Bienvenida de Wordpress

Ahora introducimos los datos que se nos solicitan, que son, el nombre de la BBDD, el nombre del usuario de la BBDD y la password, además del servidor de donde está alojada. Una vez hecho eso nos pedirá los datos de la web en cuestión, datos como el título, el usuario administrador, y el mail del creador. Una vez tenemos todo eso ya tenemos Wordpress instalado como vemos en la imagen.

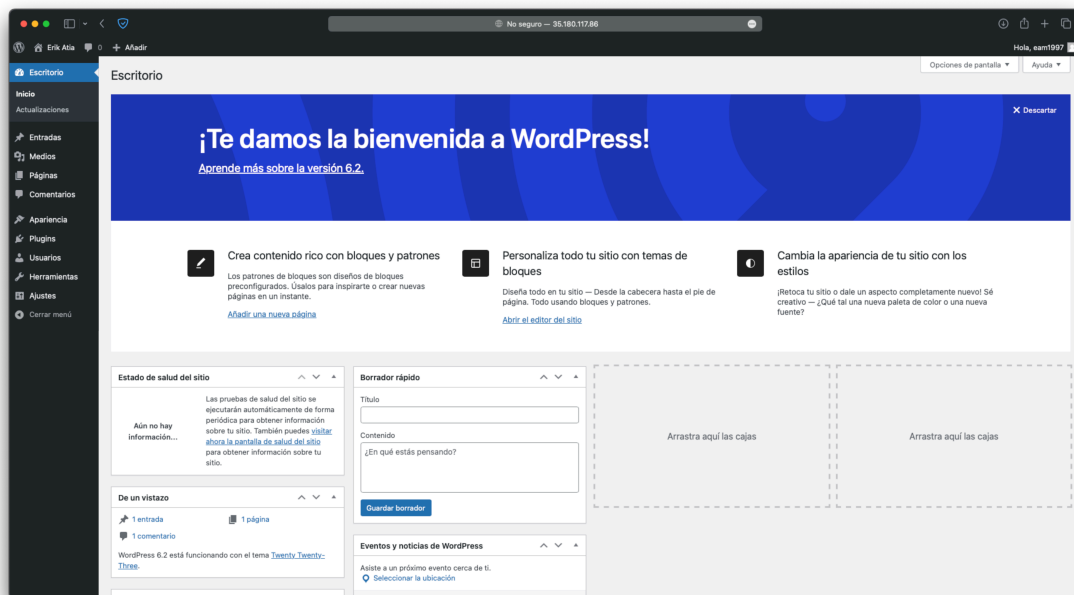


Figura 44: Inicio de Wordpress

6.6. Difusión del Podcast

En la sección de Podcasting dentro de iVoox tenemos la opción de generar el enlace Feed RSS tal como vemos en la imagen.

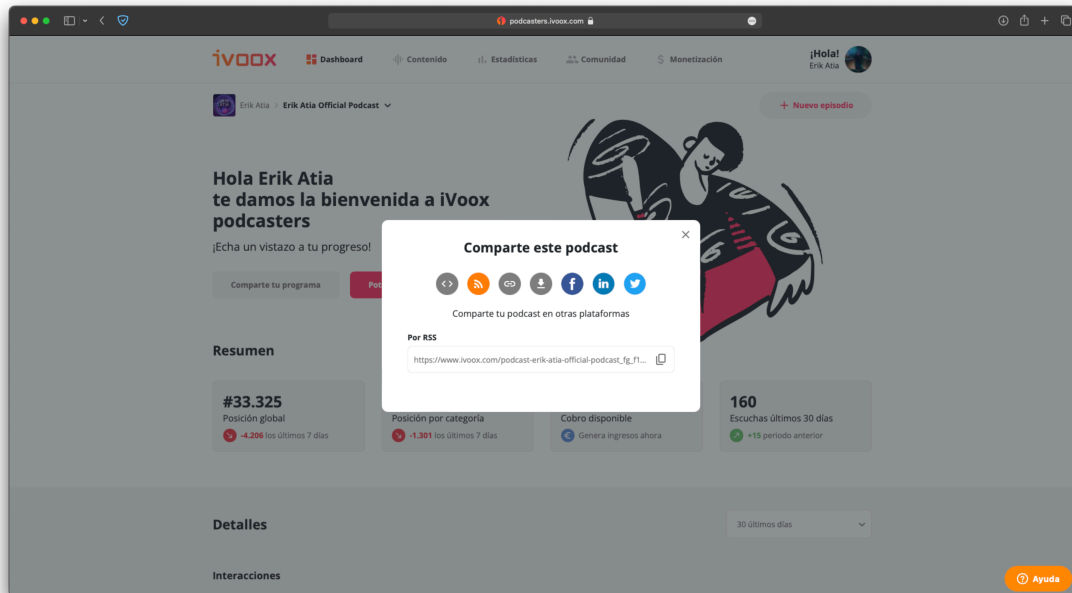


Figura 45: Feed RSS

Con esa información es muy sencillo, debemos irnos registrando en las principales plataformas y a la hora de decirles donde está alojado nuestro podcast, solo deberemos introducirles la URL RSS que iVoox nos ha generado. Voy a adjuntar una captura de Apple Podcasts Connect, donde muestro como ahora el Podcast está en la plataforma oficial de Apple disponible.

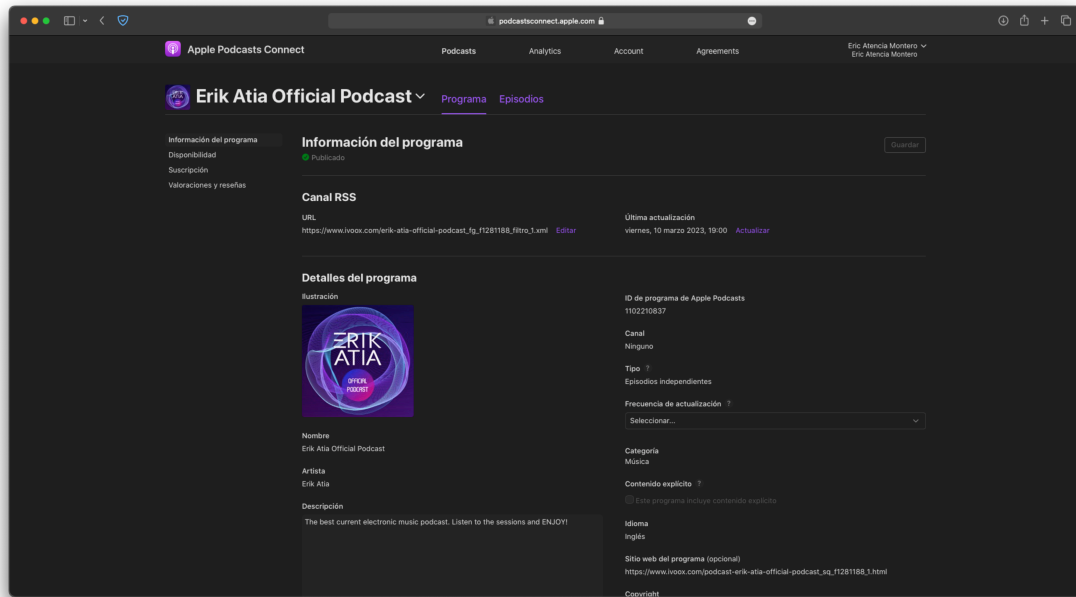


Figura 46: Apple Podcasts Connect

Este mismo proceso debemos seguirlo en el resto de plataformas interesantes, voy a adjuntar las capturas de las plataformas donde ahora está disponible.

Para finalizar, añadir que Spotify sería una de las principales plataformas que deberíamos tener en cuenta, pero por su política de derechos de autor con la música no podemos alojar nuestro podcast dentro de la sección de podcasting de su plataforma.

6.7. Certificado SSL

Tras haber generado los certificados en la plataforma mencionada anteriormente en el capítulo de materiales y recursos era necesario introducirlo en el servidor y posteriormente configurar Apache para su uso. Yo los subí al servidor mediante FTP al directorio certificates que yo mismo cree. Adjunto captura de los certificados.

```
ubuntu@srvErikAtia:~$ ls /var/www/html/certificates/  
ca_bundle.crt  certificate.crt  private.key
```

Figura 47: Certificados

Posteriormente configuré en el fichero de configuración de Apache para que apuntase a esos certificados y para que el tráfico http lo redirigiese a https.

```
GNU nano 6.2 /etc/apache2/sites-enabled/000-default.conf  
<VirtualHost *:80>  
    RewriteEngine On  
    RewriteCond %{HTTPS} !=on  
    RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R=301,L]  
</VirtualHost>  
<VirtualHost *:443>  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/html  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
    SSLEngine on  
    SSLCertificateFile /var/www/html/certificates/certificate.crt  
    SSLCertificateKeyFile /var/www/html/certificates/private.key  
    SSLCertificateChainFile /var/www/html/certificates/ca_bundle.crt  
</VirtualHost>
```

Figura 48: Fichero configuración Apache

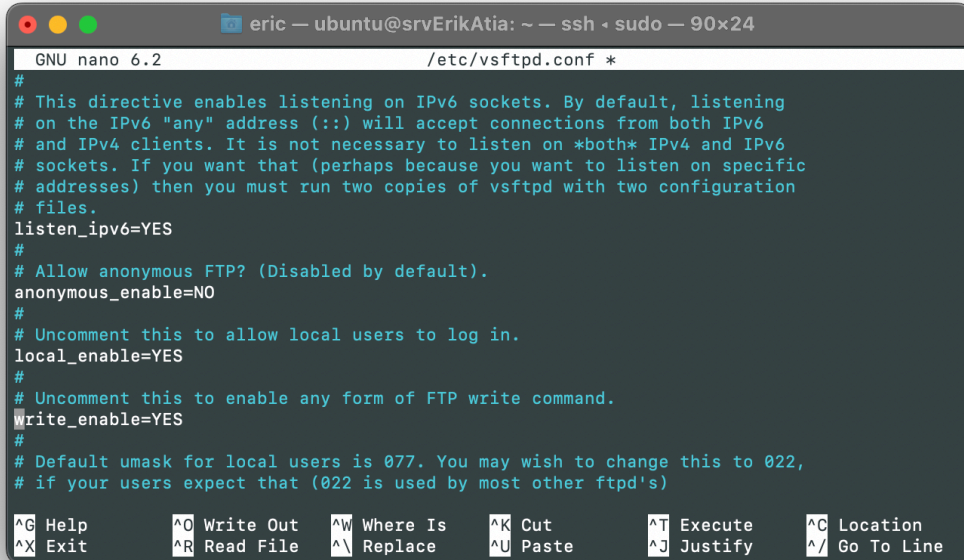
6.8. FTP

Para poder gestionar de manera comoda el servidor web habilité el FTP, primero instalando el paquete correspondiente con la siguiente orden.

```
ubuntu@srvErikAtia:~$ sudo apt-get install vsftpd  
Reading package lists... Done
```

Figura 49: Instalación VSFTPD

Posteriormente fue necesario habilitar en el fichero de configuración que fuera posible el permiso de escritura mediante el servicio, esto se hace descomentando la línea de la siguiente imagen.



```

GNU nano 6.2 /etc/vsftpd.conf *
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute   ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify   ^_ Go To Line
  
```

Figura 50: Configuración VSFTP

Después fue necesario crear un usuario para poder gestionarlo, esto se hace con el comando `useradd nombredeusuario`. Luego debí asignarle como propietario de la carpeta de la web así que lancé el comando `sudo chown -R eric:www-data /var/www/html`, con esto logramos cambiar el propietario y el grupo de un directorio recursivamente, además le asigné los permisos adecuados con `sudo chmod -R 775 /var/www/html`.

6.9. Creación Web

Para entender bien el constructor visual de DIVI es importante tener claro que siempre funciona con 3 niveles identificados por colores:

- Azul, que es digamos lo que ellos denominan sección:

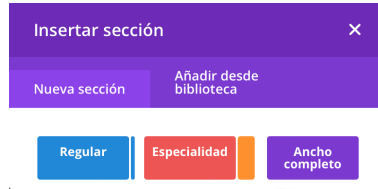


Figura 51: Sección

- Verde, define lo que ellos llaman fila, que no deja de ser la cuadrícula:



Figura 52: Fila

- Gris sería el módulo, es decir, el elemento final:



Figura 53: Modulo

6.9.1. Home

La página Home se compone de tres secciones, que son la siguientes.

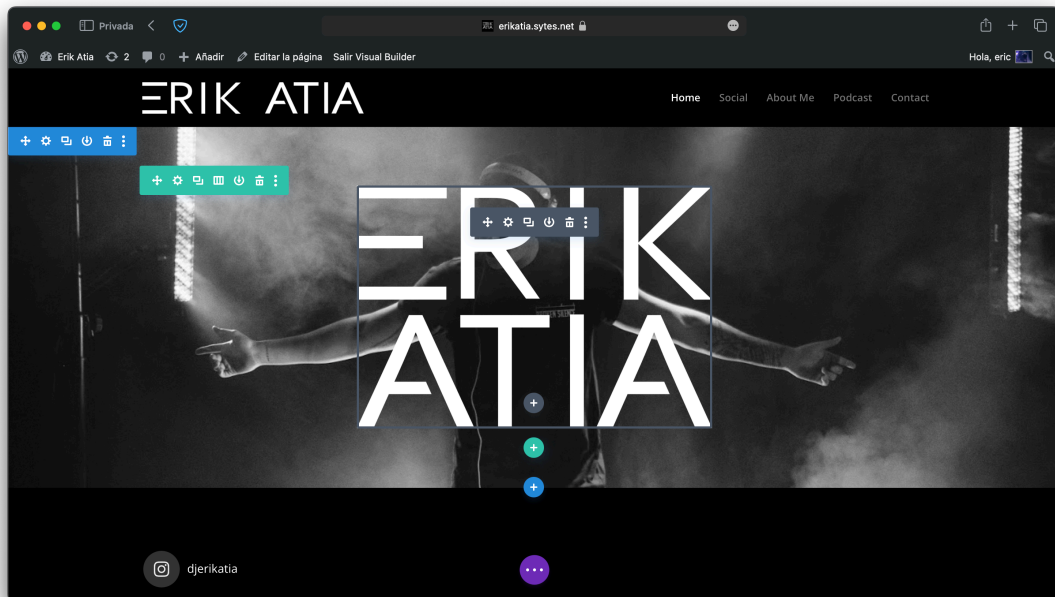


Figura 54: Home 1

En primer lugar, tenemos una sección que es a la que le he aplicado un fondo de imagen, para tener esa imagen en blanco y negro. Posteriormente en la fila la he definido como ancho completo y de módulo de nuevo he indicado que es una imagen, en este caso el logo de “Erik Atia”.

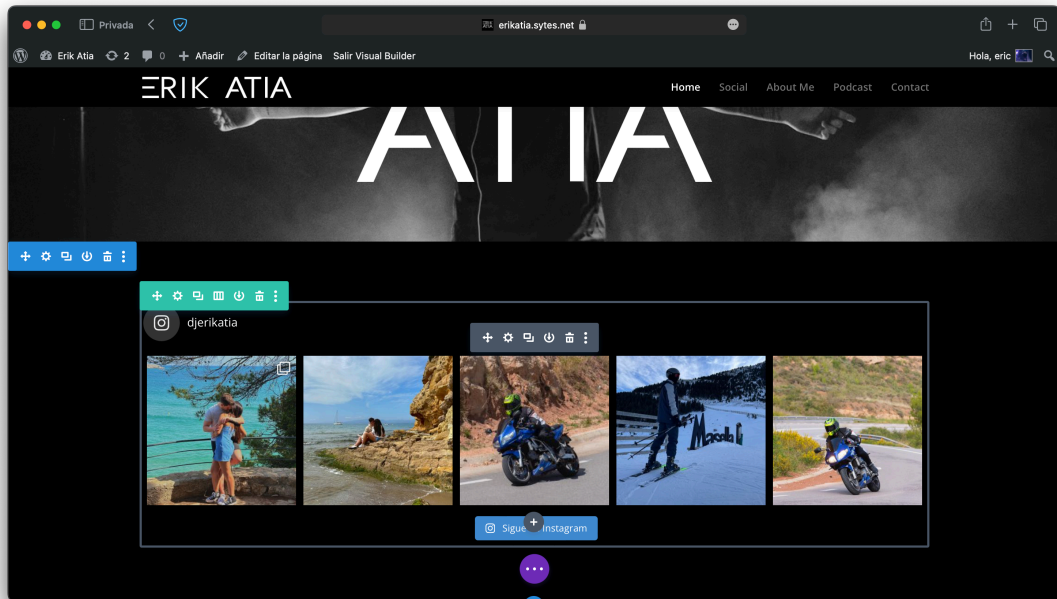


Figura 55: Home 2

Ahora tenemos la sección del feed de Instagram, es una sección con fondo negro, una fila también de ancho completo y que cuenta con el código iframe que nos facilita el plugin de Smash Ballon que hemos comentado anteriormente.

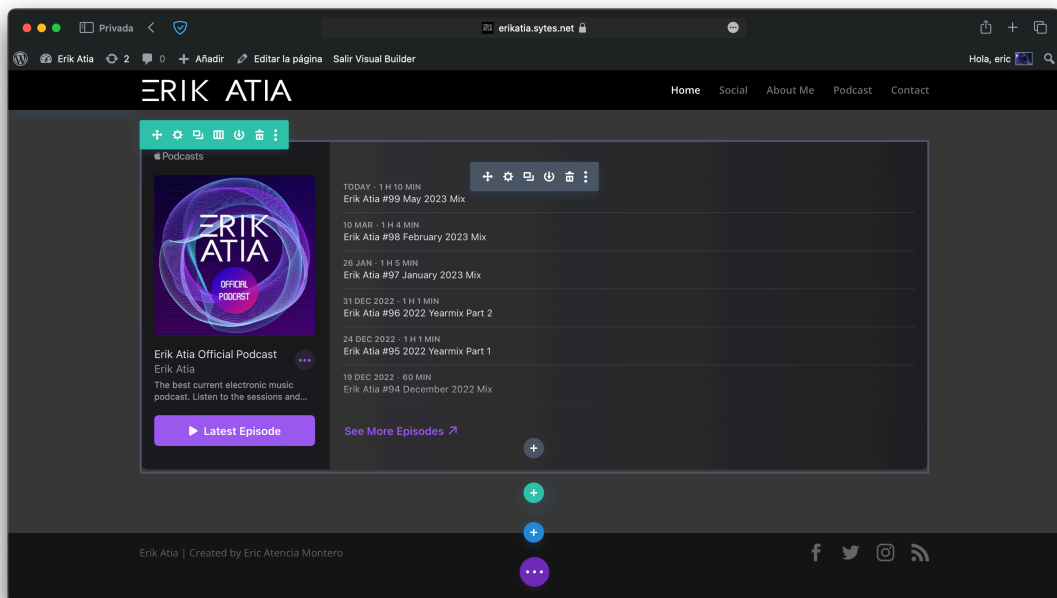


Figura 56: Home 3

Para finalizar, tenemos una última sección con fondo gris, y de ancho completo, que cuenta con el iframe que se puede generar con las herramientas de marketing de Apple. Enlace: <https://tools.applemediaservices.com/es-es/>

6.9.2. Social

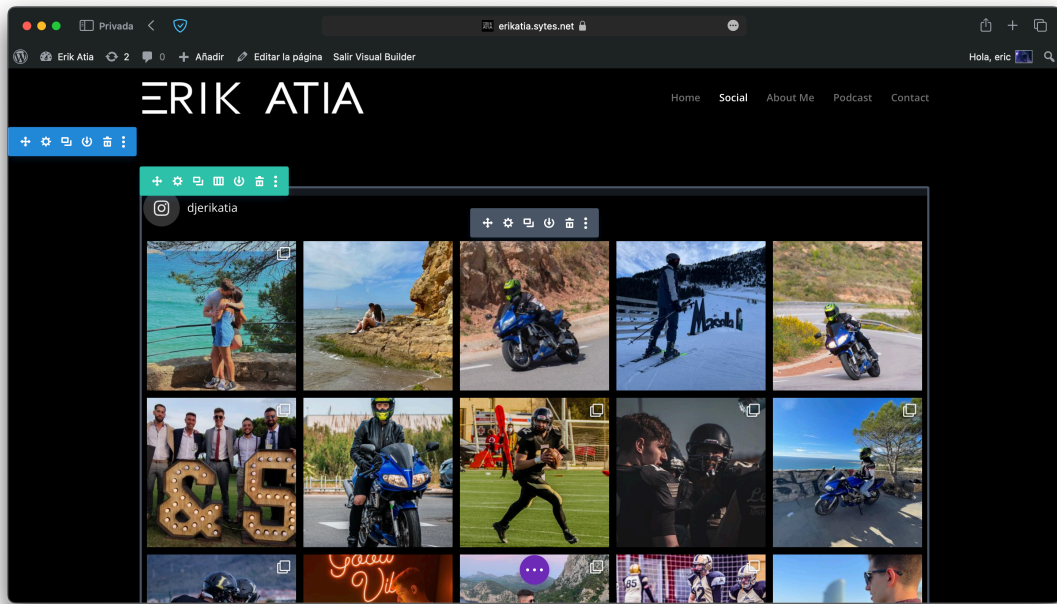


Figura 57: Social 1

Esta es más sencilla, cuenta con una única sección y con un único elemento que es otro iframe que generamos con el plugin de Smash Ballon.

6.9.3. About Me

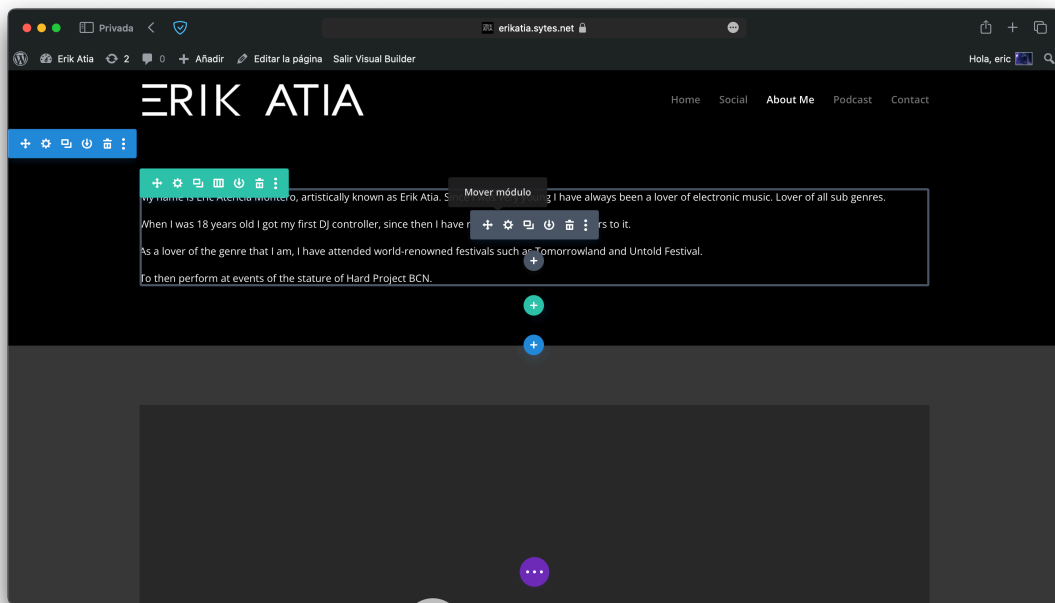


Figura 58: About Me 1

Aquí he introducido dos secciones, una para el texto, con fondo negro y ancho completo, y otra sección para el video de YouTube que incrustamos con un código que se genera desde la propia web de YouTube.

6.9.4. Podcast

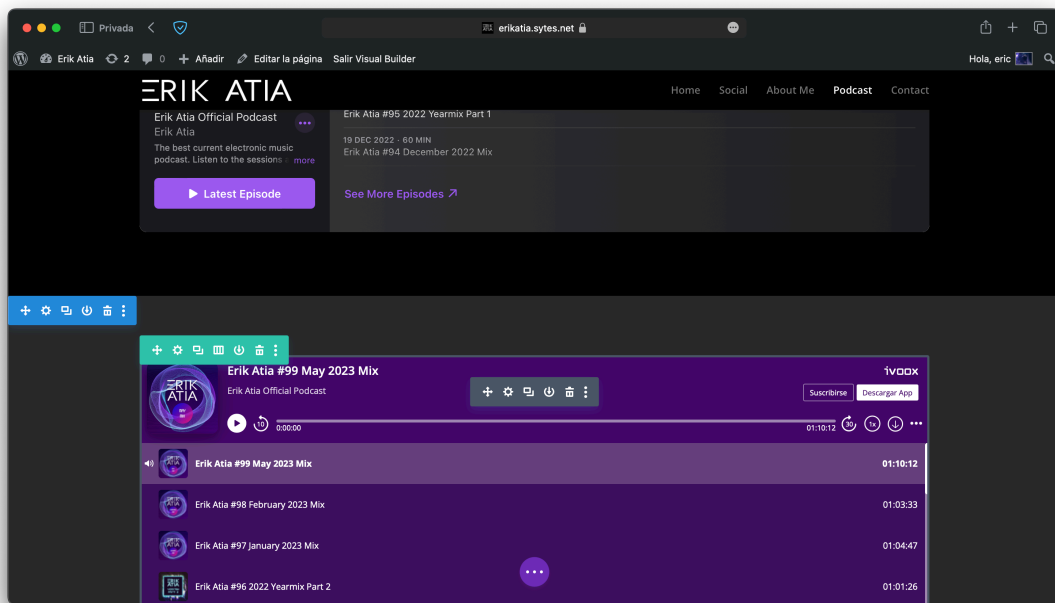


Figura 59: Podcast 1

En esta página tenemos dos secciones idénticas, entre ellas cambia el color de fondo y el iframe que utiliza fuentes distintas entre Apple Podcast e iVoox.

6.9.5. Contact

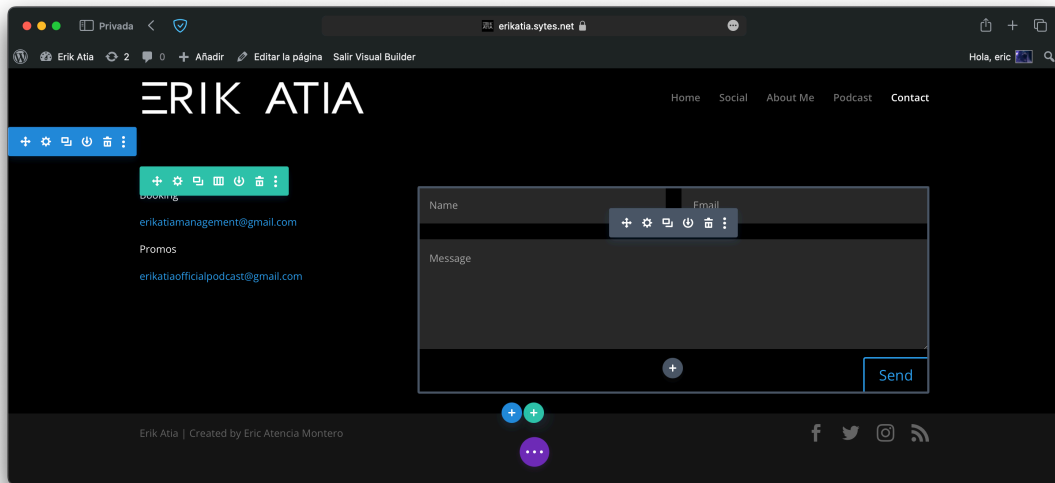


Figura 60: Contact 1

En esta página si tenemos una sección con fila de dos columnas, haciendo la de la derecha más ancha para el formulario y en la izquierda para ese mínimo de información que se necesita.

6.10. Plugins

6.10.1. Smash Ballon Instagram Feed

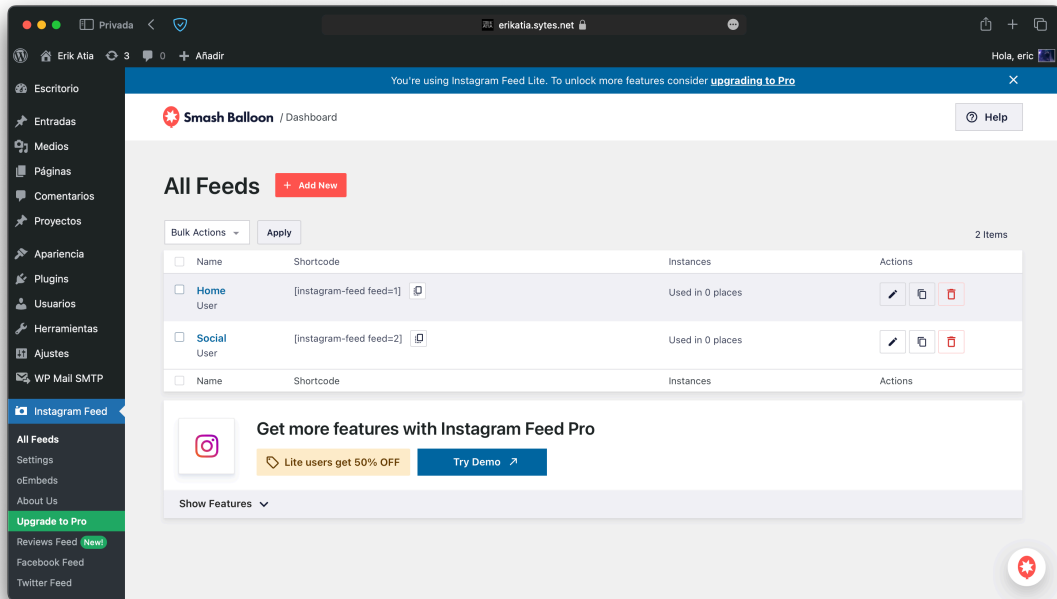


Figura 61: Smash Ballon Instagram Feed

Tal como se ve en la imagen se han creado dos feeds, uno para la página inicial y otro para la página social.

Entre ellos se diferencian por la cuadrícula, en uno es mas pequeña que en otro.

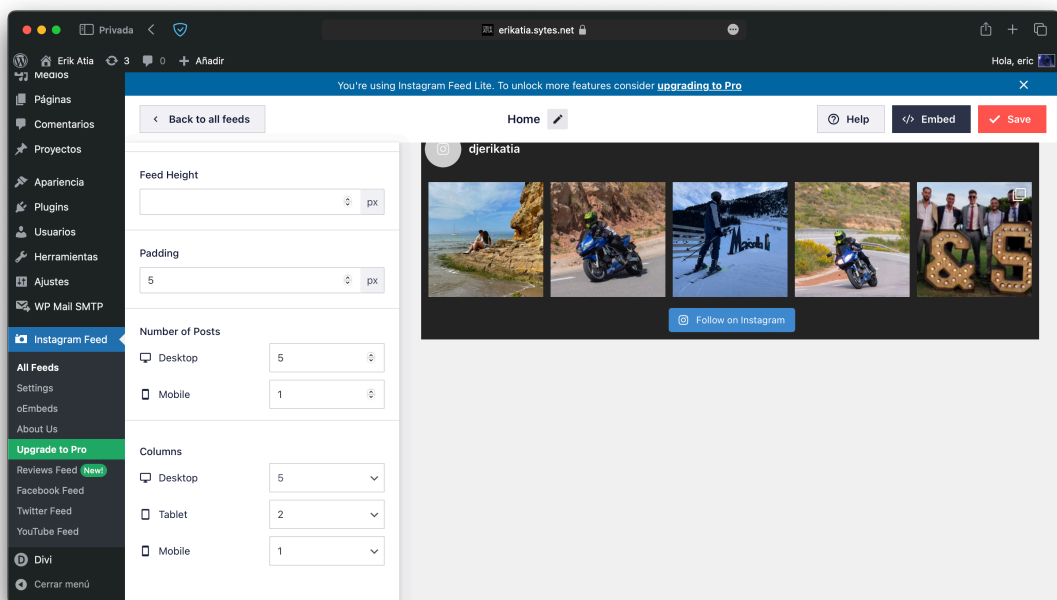


Figura 62: Feed 1

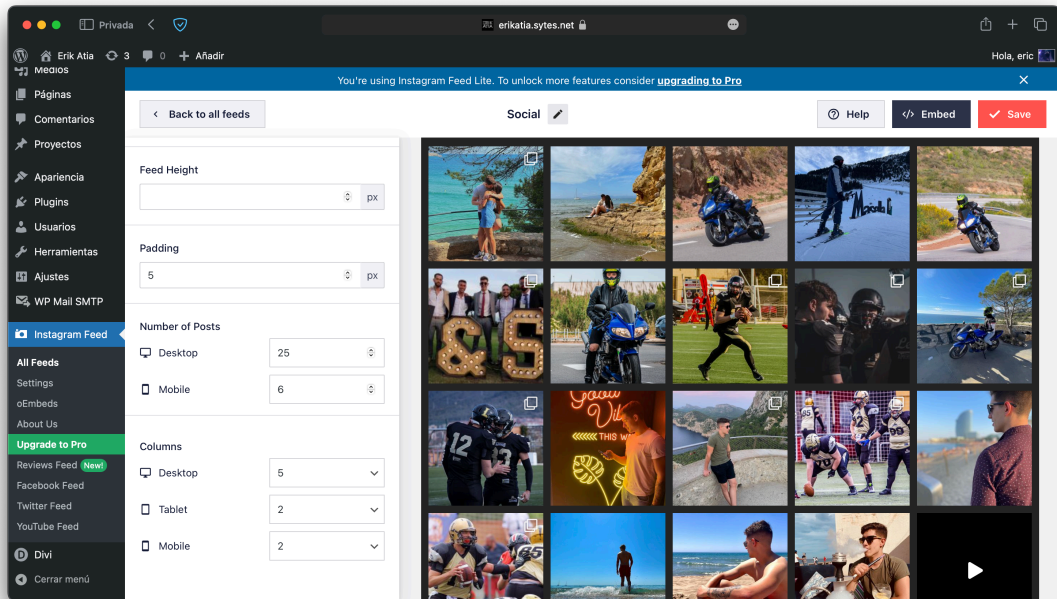


Figura 63: Feed 2

6.10.2. WP Mail SMTP

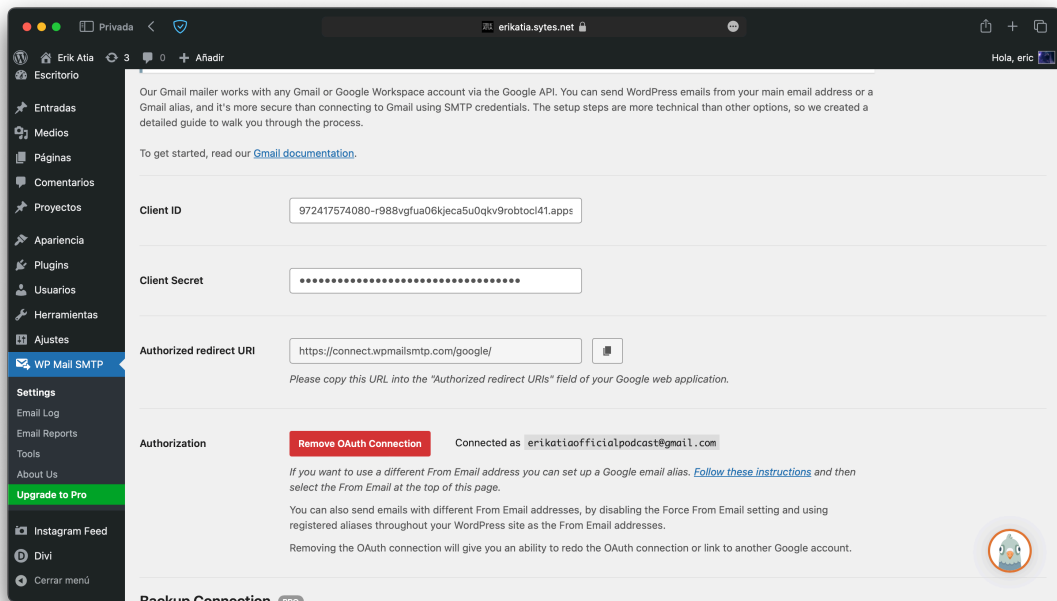


Figura 64: WP Mail SMTP

En la imagen anterior vemos como en el plugin introduzco los datos que se obtienen del apartado de desarrollo de Google.