



**UNIVERSIDAD MODELO**  
**ESCUELA DE INGENIERÍA**  
**DESARROLLO DE TECNOLOGÍA Y SOFTWARE**

**MATERIA:** Proyectos VIII  
**“IMPLEMENTACIÓN DEL MVP”**

**MAESTRA:** Ivonne Emire Cervera Cetina

**INTEGRANTES:**

Avilés Chan Alejandro  
Alpuche López Adrián  
Gamboa Bolado Rubén  
Oliva Martín Víctor

## Descripción del MVP:

Trabajando en el siguiente proyecto se desarrolló como MVP una plataforma para venta de contenido similar a sitios como Udemy, donde las personas podrán subir contenido audiovisual, con el objetivo de poder ser reconocidos en la plataforma. Podrán subir e impartir contenido, además de poder añadir imágenes o videos que consideren necesarios para sus clientes.

También tendrán la opción de una vez creada su cuenta e iniciado sesión en la plataforma, crear sus propios paquetes para ofrecerlos a los clientes interesados. Además, serán capaces de observar los contenidos más recientes en la plataforma.

## Objetivos del MVP:

Esperamos que con el MVP se logre validar la viabilidad del producto, recopilar retroalimentación del usuario, así como probar la funcionalidad de la plataforma. Cada uno de los apartados anteriores se desglosa de manera detallada en la siguiente lista:

1. **Validar la viabilidad del producto:** El principal objetivo del MVP es confirmar si la idea de la plataforma de venta de contenido educativo tipo Udemy es viable en el mercado actual. Por ende, al lanzar una versión simplificada pero funcional de la plataforma, se podrá evaluar si hay interés por parte de los usuarios y si están dispuestos a utilizarla para vender y comprar contenido y recursos educativos.
2. **Recopilar retroalimentación temprana de los usuarios:** Al proporcionar una versión inicial del producto a los usuarios, se podrá recopilar retroalimentación valiosa sobre su usabilidad real, características deseadas y posibles mejoras.

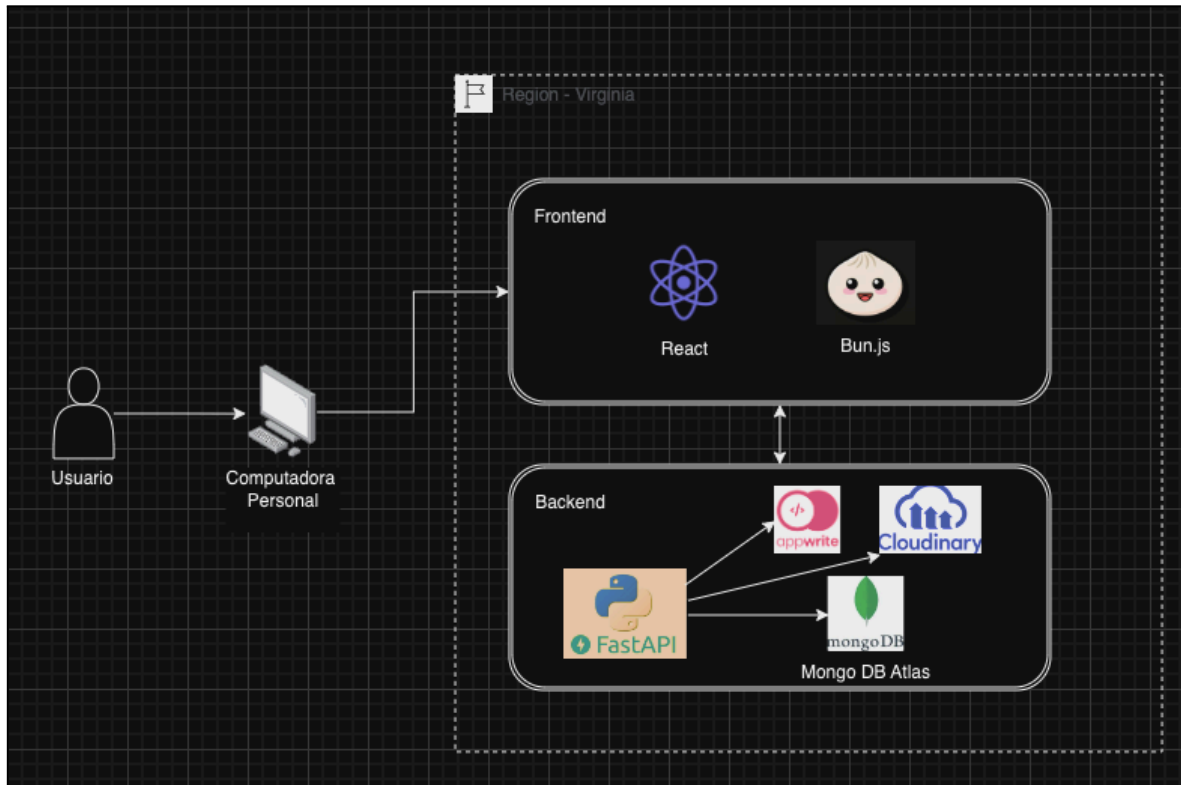
3. **Probar la funcionalidad clave de la plataforma:** El MVP se centra en implementar las características principales necesarias para permitir a los usuarios crear cuentas y subir paquetes de contenido. Al enfocarse en estos objetivos específicos, se puede determinar si la plataforma es capaz de cumplir con su propósito principal y si es suficientemente atractiva para los usuarios.
4. **Reducir el tiempo y costo de desarrollo:** Al desarrollar un MVP más simple y centrado en las características principales, se puede reducir el tiempo y los costos relacionados con el desarrollo de la plataforma. Esto nos permitirá lanzar el producto al mercado más rápidamente y con una inversión inicial muchísimo más baja, lo que minimizará el riesgo financiero y permitirá iterar y mejorar el producto de manera más ágil.

## Visión general del diseño

El diseño final del proyecto se basa en una arquitectura de software construida con React, siguiendo un enfoque de componentes reutilizables para garantizar la escalabilidad y la modularidad del sistema.

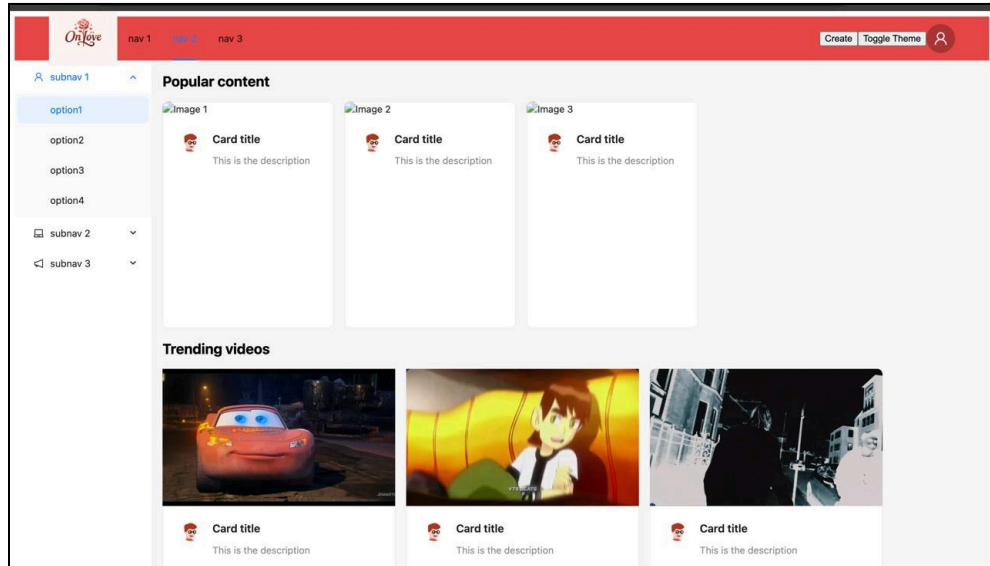
Para el almacenamiento de datos, se emplea MongoDB Atlas, ya que ofrece una flexibilidad y eficiencia en la gestión de información, además de tener una capa gratuita la cual nos permite tener la base de datos en la nube. Además de la herramienta anterior, se integra Appwrite, su servicio de autenticación de usuarios permitirá garantizar la seguridad y la gestión de usuarios.

El front end se comunicará con el backend para realizar cualquier tipo de transacción u operación, el backend está desarrollado en Python y Fast API, que se encarga de la lógica de negocio y la gestión de datos. Una característica clave de la arquitectura es la conexión en la nube con Cloudinary, un servicio de almacenamiento de archivos que facilita la gestión de videos y fotos, este servicio también es accesible solo desde el backend.

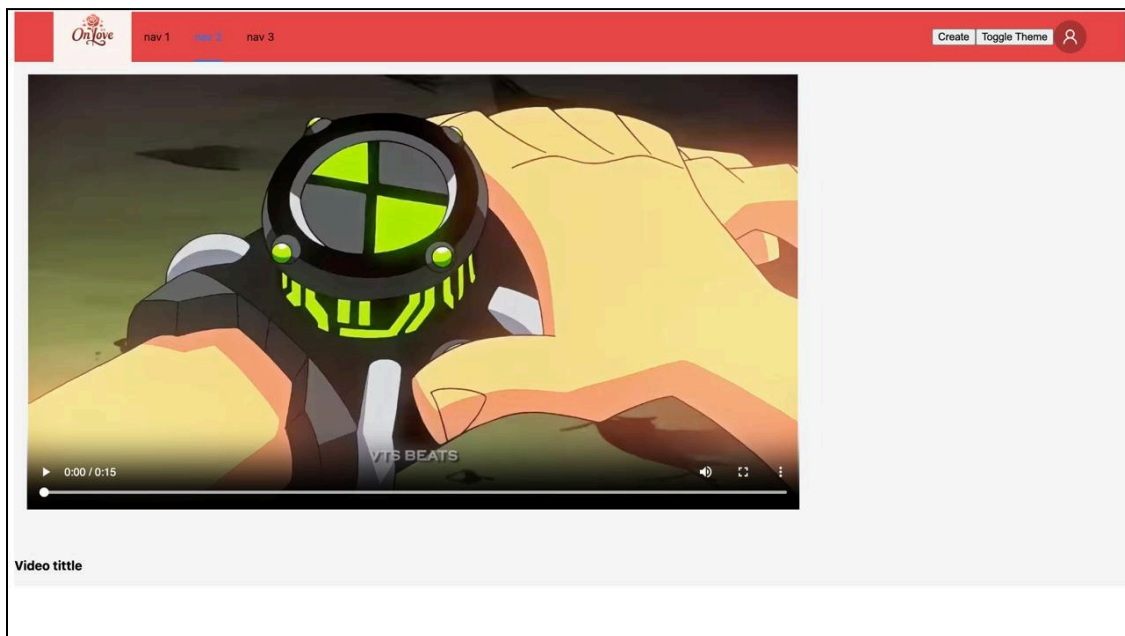


## Interfaz de usuario

La interfaz de usuario del MVP exhibe tarjetas en pantalla que representan los contenidos disponibles, destacando con una imagen el tema central del contenido. Se ha optado por una paleta de colores rojo y blanco para asegurar un contraste atractivo y fácil en la vista del usuario. En el lado izquierdo, se encuentra un menú desplegable diseñado para facilitar la navegación, permitiendo seleccionar contenidos según diferentes temas de interés.



Al seleccionar el contenido deseado, se enviará al usuario a la pantalla donde podrá ver los contenidos subidos por los usuarios. Dado que los videos son el principal enfoque, ocupan la mayor parte de la pantalla.



En la siguiente pantalla, el usuario podrá ingresar la información y los recursos necesarios para subir su contenido. Deberá introducir el título, la descripción del

video, cargar una portada, especificar el precio deseado, añadir etiquetas y, por último, cargar el video antes de pulsar el botón "Subir".

### Upload Content

Title:

Enter description

Premium

↓ Upload Cover

+

Upload

0

Price: 0

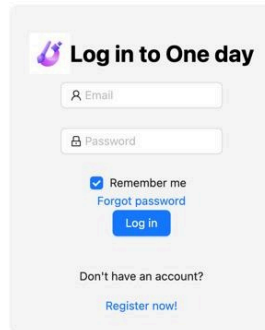
+ New Tag

Click or drag file to this area to upload

Support for a single or bulk upload. Strictly prohibited from uploading company data or other banned files.

Upload

Para el diseño del inicio de sesión, hemos adoptado un enfoque tradicional al incluir los campos de correo electrónico y contraseña, donde el usuario puede ingresar sus credenciales para acceder a su cuenta. En caso de olvidar sus credenciales, el usuario podrá recuperarlas haciendo clic en un enlace proporcionado.



**Log in to One day**

Email

Password

☒ Remember me

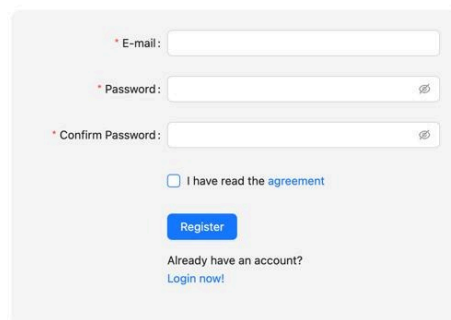
[Forgot password](#)

[Log in](#)

Don't have an account?

[Register now!](#)

En caso de que el usuario no tenga una cuenta, la página también ofrece la opción de registro. Aquí, el usuario puede ingresar su correo electrónico, contraseña y confirmar la contraseña. Si todos los campos se completan correctamente, al presionar el botón "Registrarse", la cuenta será creada en el sistema..



\* E-mail:

\* Password:

\* Confirm Password:

☐ I have read the [agreement](#)

[Register](#)

Already have an account?

[Login now!](#)

**1. Componentes React:** El frontend se estructura en componentes de React, que son bloques de construcción modulares que representan diferentes partes de la interfaz de usuario, como encabezados, tarjetas de contenido, formularios de registro, etc. Cada componente puede tener su propio estado interno y puede recibir datos a través de props.

**2. Gestión de eventos:** Los componentes de React pueden responder a eventos del usuario, como clics de botones o cambios en los campos de entrada. Estos eventos pueden desencadenar cambios en el estado de la aplicación, lo que a su vez puede provocar actualizaciones en la interfaz de usuario para reflejar los cambios.

**3. Enrutamiento:** Para una aplicación de página única (SPA), se utiliza enrutamiento para permitir la navegación entre diferentes vistas de la aplicación sin recargar la página. Bibliotecas como React Router son comunes para manejar el enrutamiento en aplicaciones React.

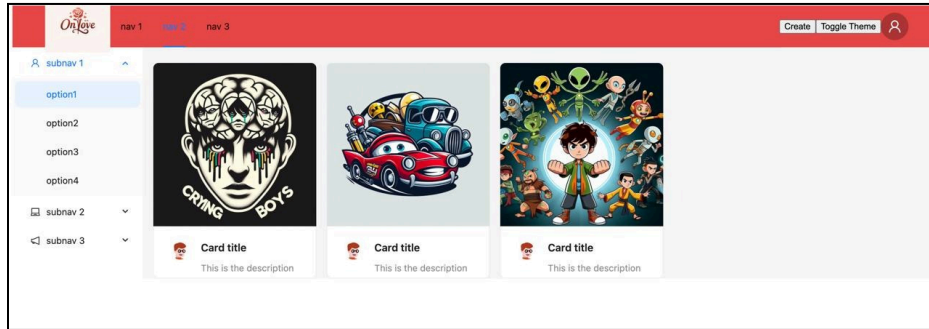
## Iteraciones y mejoras

El diseño final del MVP ha experimentado una mejora notable con respecto a las iteraciones anteriores.

### Iteración inicial

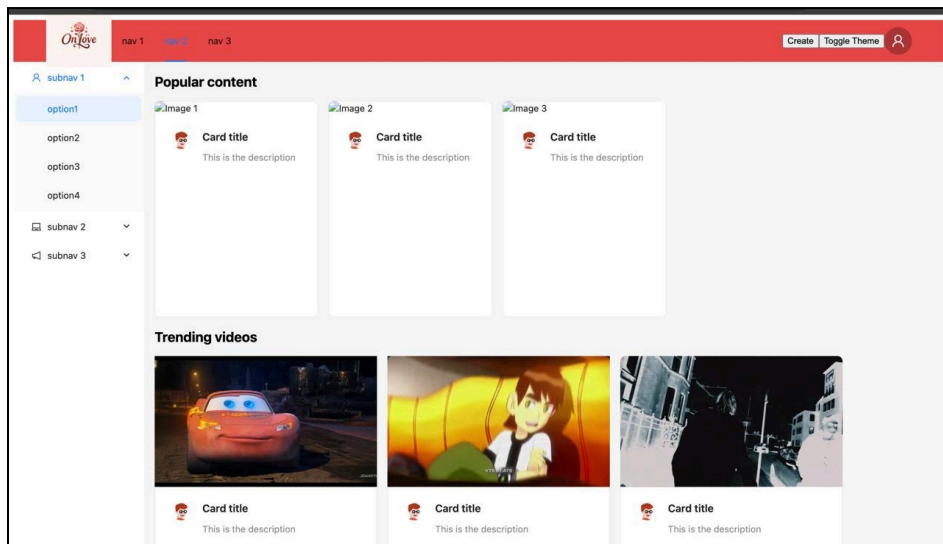
Se ha logrado una presentación más atractiva y una navegación más intuitiva en la página web, lo que facilita a los usuarios acceder fácilmente al contenido de su interés.





## Iteración actual

Además, se ha mejorado significativamente la funcionalidad del menú lateral y la disposición general de los elementos en la interfaz. Se han implementado mejoras adicionales, como la optimización del espacio, para ofrecer una experiencia aún más fluida y agradable para los usuarios.



Además de estos avances, el tiempo de respuesta de la página ha sido considerablemente optimizado. Basándonos en la retroalimentación recibida por parte de nuestros usuarios, nos hemos dedicado a mejorar este aspecto específico, lo que ha resultado en una experiencia de navegación más rápida y eficiente en la interfaz.

## Estrategia de pruebas

Para garantizar la calidad y la funcionalidad del MVP, se implementaron estrategias y pruebas exhaustivas que abarcaron varios aspectos del desarrollo del producto. A continuación, se presenta la lista de las pruebas realizadas.

**1. Pruebas de unidad:** Se llevaron a cabo pruebas de unidad en cada componente React de forma individual para garantizar que funcionaran correctamente según lo previsto. Esto implicó verificar que el estado interno de los componentes fuera correcto y que respondiera adecuadamente a las acciones del usuario.

**2. Pruebas de integración:** Se hicieron pruebas de integración para asegurar que todos los componentes de la aplicación funcionaran en conjunto correctamente. Esto incluyó la validación de la comunicación entre el frontend y el backend, así como la integración fluida con servicios externos como lo son Appwrite y Cloudinary.

**3. Pruebas de aceptación del usuario:** Realizamos pruebas de aceptación del usuario para poder evaluar la experiencia general del usuario en la plataforma. Esto incluyó desde la navegación a través de la interfaz, la creación de cuentas de usuario, la subida de contenido, hasta la compra de contenido. Se recopiló la retroalimentación de los usuarios para identificar áreas de mejora.

## Resultados de las pruebas:

Gracias a las estrategias implementadas para probar el MVP de nuestra aplicación, pudimos identificar tanto aspectos positivos como áreas de mejora en el trabajo realizado. En la siguiente sección, destacaremos los aspectos positivos y las áreas que necesitan mejoras que surgieron de las pruebas realizadas.

### Aspectos positivos:

- Los componentes individuales pasaron las pruebas de unidad con éxito, demostrando su funcionalidad y robustez.
- Las pruebas de integración confirmaron que la comunicación entre el frontend y el backend funcionaba según lo previsto, garantizando una experiencia sin interrupciones para los usuarios.
- Las pruebas de aceptación del usuario revelaron que la navegación por la plataforma era intuitiva y que las funciones principales, como la subida de contenido y la creación de cuentas, eran intuitivas y fáciles de usar.

### **Áreas de mejora identificadas**

- Se encontraron algunos problemas de rendimiento durante las pruebas de carga, especialmente al cargar grandes cantidades de contenido multimedia. Se implementaron medidas para optimizar el rendimiento y mejorar los tiempos de carga.
- Se identificaron algunos errores de validación en los formularios de registro y subida de contenido. Estos problemas fueron abordados mediante la mejora de la validación del lado del cliente y del servidor.
- La interfaz no es muy agradable para los usuarios, por lo que se tiene pensado rediseñar la interfaz para versiones futuras.

### **Cumplimiento de objetivos:**

Los logros obtenidos a través del MVP es que se ha validado con éxito la viabilidad del producto al demostrar interés por parte de los usuarios en la plataforma de venta de contenido educativo. La respuesta inicial de los usuarios proporciona una indicación positiva de la demanda en el mercado para este tipo de servicio.

Además, se ha recopilado retroalimentación valiosa de los usuarios durante la fase de MVP, lo que nos permitió identificar áreas de mejora y características deseadas. Esta retroalimentación se utilizará para guiar el desarrollo futuro de la plataforma y asegurar que satisfaga las necesidades y expectativas de los usuarios de manera más efectiva. El MVP se ha centrado en implementar las características principales necesarias para permitir a los usuarios crear cuentas, subir contenido y crear paquetes de contenido. Estas funciones clave se han probado con éxito, demostrando que la plataforma es capaz de cumplir con su propósito principal y proporcionar una experiencia satisfactoria para los usuarios.

Por último, hemos descubierto el área de mejora, siendo la reducción significativa en el tiempo y los costos asociados con el desarrollo de la plataforma. Esto nos permitirá lanzar el producto al mercado más rápidamente y con una inversión inicial más baja, minimizando el riesgo financiero y facilitando la iteración y mejora continua del producto en base a la retroalimentación recibida.

## **Métricas de éxito:**

El éxito obtenido se ha observado en una tasa de retención de usuarios favorable durante la fase de MVP, lo que indica que los usuarios encuentran valor en la plataforma y continúan utilizando sus servicios después de registrarse.

Se han recopilado comentarios positivos de los usuarios durante la fase de MVP, lo que sugiere un alto nivel de satisfacción con la experiencia general de la plataforma. Esto nos permitirá lanzar el producto al mercado más rápidamente, capitalizando la demanda existente y ganando una ventaja competitiva en el espacio de la venta de contenido educativo en línea.

## **Documentación final:**

La documentación del cierre del proyecto ha sido vital para capturar y preservar los detalles del MVP implementado, ya que obtuvimos aprendizajes y recomendaciones para futuras iteraciones. Proporcionándonos un registro histórico que puede ser utilizado para referencia, así como para informar y guiar proyectos similares en el futuro.

Los detalles del MVP implementado se han centrado en desarrollar una plataforma para la venta de contenido educativo similar a Udemy. Permite a los usuarios subir, impartir y vender su propio contenido, además de ofrecer la opción de crear paquetes de contenido para los clientes interesados. Se ha utilizado una arquitectura de software basada en React y JavaScript para el frontend, con MongoDB para el almacenamiento de datos, Appwrite para la autenticación de usuarios y Cloudinary para el almacenamiento de contenido multimedia.

Durante la fase de implementación, se obtuvieron varios aprendizajes valiosos. Se confirmó la viabilidad del producto mediante la validación de concepto y la recopilación de retroalimentación temprana de los usuarios. Se demostró que la plataforma es capaz de cumplir con sus funciones principales y de proporcionar una experiencia satisfactoria. Se identificaron áreas de mejora, como la optimización del rendimiento y la mejora de la validación de formularios, que se abordarán en futuras iteraciones.

## **Lecciones aprendidas:**

Una de las lecciones aprendidas durante todo el proceso de implementación, fue la importancia que tiene la iteración y mejora continua en un proyecto de este tipo. Al lanzar una versión simplificada pero funcional de la plataforma pudimos aprender cómo la retroalimentación valiosa de los usuarios y utilizarla para guiar el desarrollo futuro, son importantes para la planificación y la gestión de recursos,

especialmente en términos de tiempo y costos asociados con el desarrollo del proyecto.

Otra lección fue que hemos aprendido la importancia de la comunicación y la colaboración entre los miembros del equipo, por que, la coordinación efectiva entre el equipo de desarrollo como pueden ser: el equipo de diseño y de desarrollo, ha sido fundamental para tener éxito. También hemos aprendido la importancia de la flexibilidad y la adaptabilidad en un entorno de desarrollo ágil, donde los requisitos y las prioridades pueden cambiar rápidamente.