

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“APLICACIÓN WEB PARA LA DIFUSIÓN Y PRESERVACIÓN
DEL LEGADO DE LA FERIA DE SOFTWARE EN LA
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA”

JUAN ANDRÉS MIRA GRISALES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Luis Hevia Rodríguez
Profesor Correferente: Cecilia Reyes Covarrubias

Noviembre - 2024

DEDICATORIA

En dedicatoria a mi familia y a todos mis cercanos que fueron partes de este gran proceso, apoyándome día a día con ánimo para seguir adelante y más en esos días grises donde todo era más difícil.

AGRADECIMIENTOS

Quiero agradecer a mi familia: a mi mamá, mi tía, mis abuelitos y a mi pequeño hermano Mateito, por apoyarme en cada etapa de este largo e intenso camino universitario, por estar siempre presentes y acompañarme en todo lo que necesité.

También quiero dar las gracias a todas las personas que conocí en la universidad. En especial a Marco y Javier, con quienes pasé la mayoría del tiempo estudiando y disfrutando todo el proceso, y de quienes me llevo una amistad para toda la vida; incluso tuvimos la oportunidad junto a Marco de hacer nuestra práctica profesional en Buk. A mis amigos Rafael y Felipe, con quienes creamos CuisineAR para la Feria de Software y logramos ganar un premio juntos.

Por último, y no menos importante, agradezco a todos los profesores que dedicaron su tiempo a enseñarnos y acompañarnos en este camino. Al Profesor Luis Hevia, por recibirnos de manera cercana y permitirnos trabajar junto a él en muchas tareas del Departamento de Informática. A la Profesora Cecilia Reyes, por su dedicación y cariño al enseñarnos. A la Profesora Andrea Vásquez, quien fue la primera persona en enseñarme a programar y me hizo dar cuenta de que era algo que realmente disfrutaba. Y al Profesor Mauricio Figueroa, quien estuvo presente en la última parte de mi carrera con los ramos electivos, y me mostró que aprender no tiene que ser algo complicado, sino que con ganas y disposición se pueden lograr muchas cosas.

RESUMEN

Resumen— La "Feria de Software" de la Universidad Técnica Federico Santa María, con más de 30 años de historia, ha sido un evento emblemático en la formación de los estudiantes de informática, sin embargo, la falta de un mecanismo para gestionar la información generada ha dificultado la preservación de su legado. En esta memoria se detalla el desarrollo de una aplicación web diseñada para centralizar y organizar los contenidos claves de la Feria de Software. Esta aplicación tiene como objetivo no solo preservar esta información, sino también difundirla y garantizar su acceso para futuras generaciones, haciendo uso de tecnologías modernas y metodologías que se adaptan a este contexto específico.

Palabras Clave— Aplicación web; Desarrollo de Software; Preservación; Legado Histórico; Difusión.

ABSTRACT

Abstract— The "Feria de Software" of the Universidad Técnica Federico Santa María, with over 30 years of history, has been an emblematic event in the education of computer science students. Despite this, the lack of a mechanism to manage the generated information has hindered the preservation of its legacy. This report details the development of a web application designed to centralize and organize the key content of the Feria de Software. This application aims not only to preserve this information but also to disseminate it and ensure its access for future generations, utilizing modern technologies and methodologies that adapt to this specific context.

Keywords— Web Application; Software Development; Preservation; Historical Legacy; Dissemination.

GLOSARIO

API: Interfaz de Programación de Aplicaciones.

CCSDS: Council of the Consultative Committee for Space Data Systems.

DI: Departamento de Informática.

DO: Data Object.

FESW: Feria de Software.

IA: Inteligencia Artificial.

ISO: Organización Internacional de Normalización.

IP: Information Package.

OAIS: Open Archival Information System.

ORM: Object-Relational Mapping.

PDI: Preservation Description Information.

SUS: System Usability Scale.

UTFSM: Universidad Técnica Federico Santa María.

WYSIWYG: What You See Is What You Get.

ÍNDICE DE CONTENIDOS

RESUMEN	IV
ABSTRACT	IV
GLOSARIO	V
ÍNDICE DE FIGURAS	IX
ÍNDICE DE TABLAS	X
INTRODUCCIÓN	1
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	2
1.1 Contexto y Situación Actual	2
1.2 Definición del Problema	3
1.3 Objetivos	6
1.3.1 Objetivo General	6
1.3.2 Objetivos Específicos	6
CAPÍTULO 2: MARCO CONCEPTUAL	7
2.1 Preservación Digital	7
2.1.1 Contexto Histórico	7
2.1.2 Era Digital	7
2.2 Desafíos de la Preservación Digital	8
2.3 Ejemplos de Preservación y Difusión del Legado Histórico en Plataformas Di- giales	10
2.3.1 FC Barcelona	10
2.3.2 Real Madrid CF	11
2.4 Modelo OAIS	11
2.4.1 Roles en OAIS	12
2.4.2 Tipos de Paquetes	13
2.4.3 Tipos de Comunicación	14
2.5 Content Managment System (CMS)	15
2.6 Front-end	17

2.6.1	Vue JS	17
2.6.2	Vuetify	17
2.6.3	Axios	17
2.7	Back-end	17
2.7.1	API Rest	18
2.7.2	Express	18
2.7.3	Object-Relational Mapping (ORM)	18
2.8	Object Storage	19
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN		20
3.1	Levantamiento de Requerimientos	20
3.1.1	Alcance del Proyecto	20
3.1.2	Marco del Proyecto	21
3.1.3	Tipos de Contenidos	21
3.2	Diseño Modelo de Datos	23
3.3	Diseño Paquete de Información	25
3.4	Diseño API	25
3.4.1	Arquitectura	26
3.4.2	Seguridad	28
3.4.3	Endpoints	29
3.5	Almacenamiento de Datos y Archivos	38
3.5.1	Base de Datos PostgreSQL	38
3.5.2	MinIO Object Store	39
3.6	Diseño Interfaz	39
3.6.1	Componentes Transversales	40
3.6.2	Vistas Generales	45
3.6.3	Vistas Administración	58
CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN		61
4.1	Digitalización, Recopilación y Almacenamiento de Contenidos	61
4.2	Pruebas de Usuario	62
4.3	Ajustes en la Aplicación	67
CAPÍTULO 5: CONCLUSIONES		68
5.1	Conclusiones Generales	68

5.2	Cumplimiento de los objetivos	69
5.3	Trabajo Futuro	70
5.4	Palabras finales del autor	71
REFERENCIAS BIBLIOGRÁFICAS		73

ÍNDICE DE FIGURAS

1	Estructura organizacional Feria de software.	3
2	Árbol del problema.	5
3	Web FC Barcelona.	10
4	Web Real Madrid CF.	11
5	Entorno de OAIS.	13
6	Information Package (IP).	13
7	Diagrama de flujo de datos OAIS.	15
8	Modelo de Datos	24
9	Capas internas API	27
10	Menú de Navegación	41
11	Menú de Navegación versión mobile	41
12	Componente FeriaSelectGlobal	42
13	Componente AdminOptions	43
14	Componente FileUploader	44
15	Componente Gallery	45
16	Vista Inicio	46
17	Componente FeriaCard	47
18	Vista Hitos Históricos	48
19	Componente HitosHistoricosCard	49
20	Vista Hitos Históricos	49
21	Componente PreservacionCard	50
22	Vista Galerías	51
23	Componente GaleriaCard	52
24	Vista Investigaciones	52
25	Componente InvestigacionesCard	53
26	Vista Proyectos Destacados	54
27	Componente ProyectosDestacadosCard	55
28	Vista Impacto en los Medios	55
29	Componente ImpactoMediosCard	56
30	Vista Testimonios	57
31	Componente TestimoniosCard	58
32	Vista Administrador	59

33	Tabla de Información vista Administración	60
34	Respuesta a ChatGPT	71

ÍNDICE DE TABLAS

1	Resultados System Usability Scale.	66
---	--	----

INTRODUCCIÓN

La Feria de Software nace como una innovadora propuesta que busca potenciar la formación profesional de los futuros informáticos, desde 1992 este evento lleva generando un gran impacto en la comunidad estudiantil y profesional, sirviendo como un espacio para crear soluciones tecnológicas que abordan problemáticas reales, por ello es fundamental preservar su legado a lo largo del tiempo.

Actualmente el Departamento de Informática no tiene una metodología o mecanismo para poder abordar la gran cantidad de información que se genera año a año en cada una de las ediciones de la Feria de Software, por lo que es fundamental plantear una solución para abordar esta problemática.

En esta memoria se presenta el contexto y situación actual de la Feria de Software, dando a conocer su problemática en términos de gestionar el gran volumen de información que genera, se presenta el estado del arte y cómo otras organizaciones han podido solventar problemáticas similares haciendo uso de tecnología y metodologías, para continuar con la propuesta de una solución que no solo busca almacenar la información, sino que también se enfoca en que la información pueda estar disponible a lo largo de tiempo y de fácil acceso. También se detalla el proceso de desarrollo, implementación a nivel técnico y los resultados obtenidos tras la validación de la aplicación.

CAPÍTULO 1

DEFINICIÓN DEL PROBLEMA

1.1. Contexto y Situación Actual

La Universidad Técnica Federico Santa María (UTFSM) se funda en la ciudad de Valparaíso, Chile, en el año 1931, desde su fundación se han generado diversos departamentos dentro de la Universidad, donde en el año 1975 fue pionera en Chile al crear el departamento de Ciencias en Computación, que posteriormente en 1980 cambiaría su nombre a como lo conocemos hoy en día como el Departamento de Informática (DI) [Departamento de Informática, 2023].

A lo largo de los años, el Departamento de Informática ha desarrollado diversas actividades y eventos, tanto en el ámbito académico como para el desarrollo de diferentes comunidades, es por ello que en el año 1992, surge una innovadora propuesta académica que busca potenciar la formación profesional de los futuros informáticos, más conocida como la Feria de Software (FESW)[Reyes Covarrubias y Hevia Rodríguez, 2018], donde se desarrollan productos de software que solucionan problemas reales de la sociedad, es aquí donde se potencia la capacidad de “aprender haciendo”, además se busca que los futuros ingenieros pueda fomentar el espíritu emprendedor y que a lo largo de la duración del proyecto, enfrenten diversos desafíos que los sumergen en un entorno muy cercano a su futuro desarrollo profesional.

La FESW surge en un comienzo por los profesores del Departamento de Informática Luis Hevia y Cecilia Reyes, con el paso del tiempo más personas se han sumado al equipo de trabajo, generando una estructura organizacional bien definida como se observa en Figura 1, logrando alcanzar hasta la fecha más de 31 ediciones.

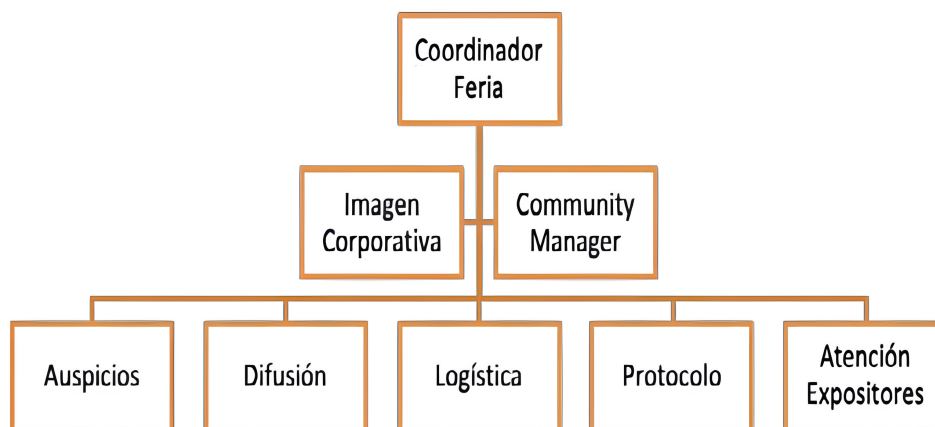


Figura 1: Estructura organizacional Feria de software.

Fuente: [Reyes Covarrubias y Hevia Rodríguez, 2018].

A partir del 2014, la Feria de Software pasa a ser parte del Ciclo de Desarrollo profesional dentro del plan de estudios, con las asignaturas de Gestión de Proyectos Informáticos y Taller de Desarrollo de Proyectos de Informática, las cuales se dictan en el último año y medio de la carrera, de esta forma se logra integrar los aprendizajes previos preparando a los estudiantes para su proceso de titulación y el mundo laboral[Departamento de Informática, 2023].

Dada la magnitud de este importantísimo evento se ha generado una gran cantidad de información de diversos tipos y formatos, tanto recursos digitales como físicos, es aquí donde surge una importante pregunta, ¿cómo se gestiona este gran volumen de información?, no solo con el objetivo de poder preservarlo sino también para su posterior difusión y de esta forma genera un valioso legado para las futuras generaciones.

1.2. Definición del Problema

A pesar del gran éxito y la relevancia de la FESW, el Departamento de Informática enfrenta una serie de desafíos importantes en la gestión de la información generada por el evento. La falta de un sistema eficaz para manejar el volumen de datos ha generado una pérdida significativa de información. Esta problemática se debe a múltiples causas, incluyendo el deterioro de los documentos con el tiempo y la ausencia de un sistema o mecanismo de almacenamiento bien definido y organizado.

El deterioro físico es una preocupación constante cuando se trata de conservar documentos impresos. Los materiales no están almacenados en condiciones óptimas, lo cual es crucial para preservar su integridad. No tener un buen sistema para guardar los documentos ha hecho que muchos se deterioren o se pierdan. Este problema se vuelve aún más serio cuando los documentos no se mantienen en condiciones adecuadas. La luz, la humedad y el tiempo pueden dañar los documentos, haciéndolos ilegibles con el tiempo. Una vez que los materiales se dañan, recuperar la información se convierte en una tarea muy complicada, y en muchos casos, la información valiosa se pierde.

Por otro lado, la información generada por la FESW no está centralizada. Está esparcida en diferentes lugares, tanto físicos como digitales, lo que hace que su gestión sea aún más difícil. La falta de una estructura clara para almacenar y organizar estos datos crea obstáculos significativos para acceder a ellos cuando se necesitan. Esta dispersión no solo dificulta la tarea de encontrar y utilizar la información, sino que también limita el acceso para quienes podrían necesitarla. El poco orden de los datos en distintos formatos y ubicaciones reduce la efectividad y el valor de la información, haciendo que no pueda ser aprovechado de buena forma.

Es fundamental abordar estos problemas con urgencia. Implementar un sistema organizado y bien estructurado para gestionar y conservar la información generada por la FESW. Solo así se podrá garantizar que la información no se pierda y que esté disponible para quienes la necesiten, tanto ahora como en el futuro, lo anterior se puede ver de una forma más resumida en la Figura 2.

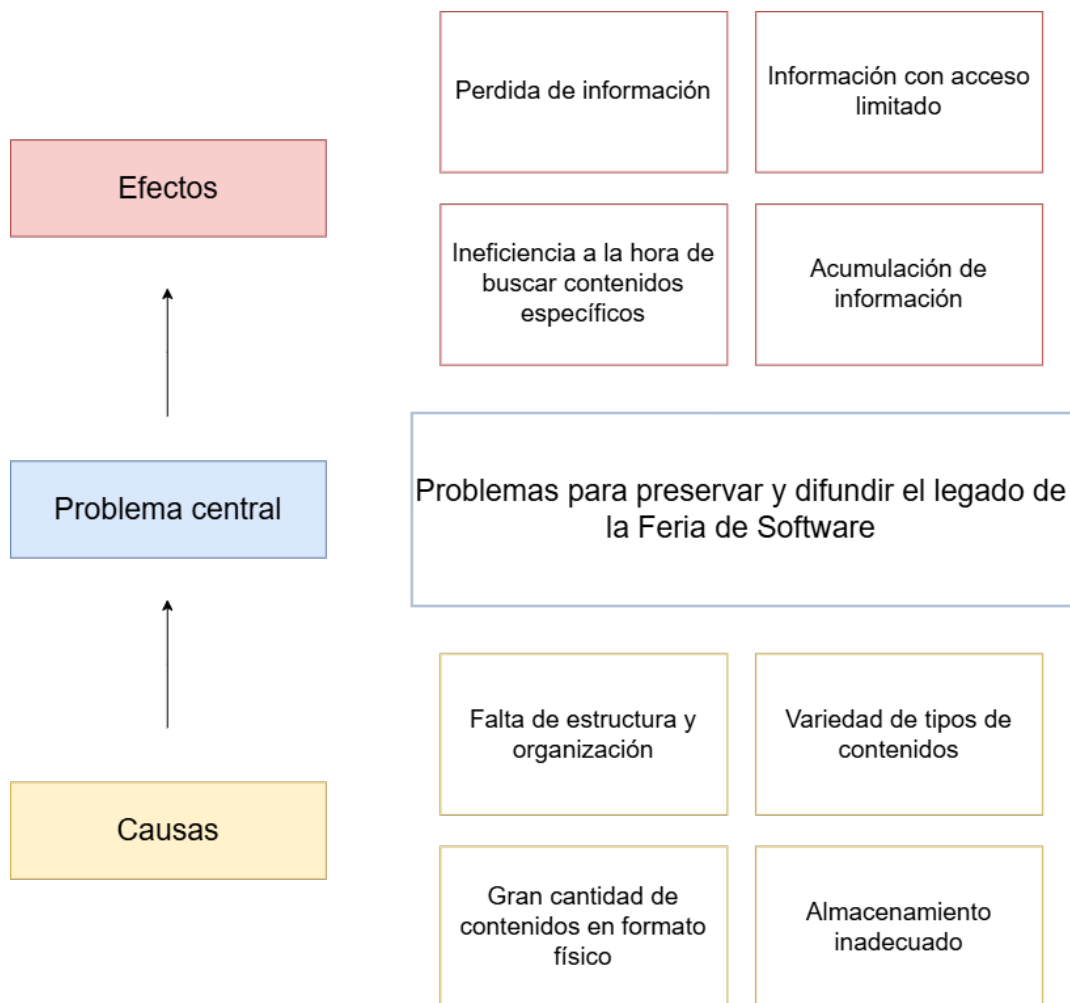


Figura 2: Árbol del problema.
Fuente: Elaboración propia.

Si se considera que la gestión de la información siga de la misma forma en que se ha estado realizando, se corre el riesgo de perder cada vez mayor cantidad de información, además de que su acceso se vuelve mucho más complicado y limitado ya que año a año el volumen de datos crece de manera apresurada, por lo que es fundamental poder dar solución a esta problemática y gestionar de una forma óptima la variedad de contenidos.

En el contexto de la era digital en la que vivimos, la accesibilidad instantánea a la información se convierte prácticamente en una necesidad. Surge así un nuevo reto: transformar los elementos físicamente almacenados, ya sea en cajas o estanterías, hacia un formato digital que permita un acceso ágil y sencillo desde cualquier parte del mundo. Este desafío abarca

la tarea de migrar recursos y datos físicos hacia un espacio digital, facilitando su disponibilidad global y su manejo en una época caracterizada por la velocidad y el fácil acceso de la información.

1.3. Objetivos

1.3.1. Objetivo General

Análisis y desarrollo de una aplicación web para la Difusión y Preservación del legado generado por la Feria de Software del Departamento de Informática en la Universidad Técnica Federico Santa María.

1.3.2. Objetivos Específicos

- Analizar la situación actual del legado de la Feria de Software y cómo se ha gestionado.
- Proponer una solución para manejar el legado de la Feria de Software.
- Implementar la solución propuesta para mantener y difundir el legado de la Feria de Software.
- Verificar el uso de la solución y realizar ajustes necesarios.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Preservación Digital

La preservación de la información es un fenómeno que ha existido siempre. Este proceso se ha manifestado a lo largo de la historia de diversas maneras, siendo fundamental para asegurar la continuidad y transmisión del conocimiento a lo largo del tiempo.

2.1.1. Contexto Histórico

Desde tiempos inmemoriales, los seres humanos han buscado formas de comunicar y conservar sus experiencias, ideas y cultura. Un ejemplo claro de esta preservación se encuentra en las pinturas rupestres, donde nuestros ancestros plasmaban sus vivencias, rituales y entorno en las paredes de las cuevas. Estas representaciones no solo eran expresiones artísticas, sino también testimonios valiosos de la vida y la sociedad de aquellos tiempos [Boté-Vericad, 2012].

A medida que las civilizaciones evolucionaron, la preservación de la información se expandió tanto en su alcance como en su representación. Las sociedades más estructuradas desarrollaron sistemas de escritura y comenzaron a registrar los acontecimientos más relevantes en papiros, pergaminos y otros medios duraderos. Este acto de registrar y documentar eventos, leyes, descubrimientos y pensamientos no solo sirvió como una forma de preservar la información, sino que también permitió compartir y transmitir ese conocimiento a generaciones futuras.

2.1.2. Era Digital

La era digital ha traído consigo una nueva forma en la que guardamos y preservamos la información [Gracy y Kahn, 2012]. En este contexto, la preservación digital hace referencia a todas aquellas técnicas y procedimientos que resultan esenciales para mantener a lo largo

del tiempo objetos digitales en un sistema de información [Boté-Vericad, 2012].

Es importante distinguir entre dos términos que suelen generar confusión:

- conservación digital
- preservación digital

La conservación digital implica la acción de mantener en el tiempo objetos digitales [Muñoz, 2006]. Mientras tanto, la preservación digital es una parte crucial de estos procesos de conservación, abarcando metodologías y procesos tecnológicos específicos cuyo propósito es garantizar el acceso continuo a esta información en el futuro [Boté-Vericad, 2012].

En un mundo donde la información digital aumenta rápidamente, mantenerla a salvo es fundamental. Los avances tecnológicos desafían constantemente nuestras habilidades para guardar y proteger estos datos, lo que nos lleva a desarrollar nuevas estrategias y métodos de manera constante. Desde sistemas de almacenamiento hasta técnicas de respaldo y organización de metadatos, todos estos aspectos son piezas fundamentales en el proceso complejo de preservar la información digital a lo largo del tiempo.

De esta forma la preservación digital no solo se trata de almacenar los datos existentes en el presente, sino que busca proyectar toda esta información a las futuras generaciones dejando un legado que perdure, de esta forma se garantiza la continuidad del conocimiento y la historia en la era digital.

2.2. Desafíos de la Preservación Digital

La preservación digital trae consigo diversos desafíos que se deben tener en cuenta, ya que afectan directamente la conservación a largo plazo de la información en formatos digitales, algunos de estos desafíos son:

- **Digitalización:** La digitalización es el proceso mediante el cual la información en formatos analógicos o físicos (como impresiones, fotografías, videos, etc.) se convierte en datos digitales. Esto implica la conversión de la información física a un formato

electrónico, permitiendo su almacenamiento, manipulación, y acceso a través de dispositivos digitales como computadoras, teléfonos inteligentes o tabletas.

- **Obsolescencia Tecnológica:** La obsolescencia tecnológica en la preservación digital es la situación en la que los medios de almacenamiento, los formatos de archivo y los dispositivos para acceder a la información están desactualizados o dejan de ser compatibles con los avances tecnológicos. Esto trae un reto importante para mantener la información a largo plazo, ya que el avance constante en la tecnología puede dificultar el acceso o la recuperación de datos almacenados en sistemas más antiguos.
- **Migración:** La migración de datos es una técnica clave para afrontar el reto de la obsolescencia tecnológica en la preservación digital. Cuando los dispositivos de almacenamiento o los formatos de archivo quedan desactualizados, la migración emerge como una solución esencial para transferir la información a sistemas más modernos y compatibles, al emplear la migración como estrategia de preservación, se busca evitar la pérdida de datos valiosos debido a la obsolescencia tecnológica.
- **Emulación:** La emulación consiste en crear un entorno virtual que reproduce sistemas informáticos antiguos. Su objetivo es mantener accesibles programas o aplicaciones que funcionaban en tecnologías obsoletas. Esto permite que los formatos y archivos antiguos puedan ser utilizados en sistemas modernos, preservando la accesibilidad que de otra manera serían inaccesibles debido al avance tecnológico. Aunque es una estrategia complementaria a la migración de datos, la emulación presenta desafíos técnicos y requiere actualizaciones constantes para su funcionamiento a largo plazo.
- **Almacenamiento:** El almacenamiento es fundamental en la preservación digital, ya que define los mecanismos técnicos utilizados, como el tipo de discos duros y el lugar de almacenamiento. Estos aspectos son críticos para lograr una preservación óptima, ya que pueden surgir problemas como corrupciones en el medio si no se gestionan y analizan adecuadamente. Es crucial establecer con precisión estos elementos para garantizar la integridad y seguridad de los datos a lo largo del tiempo.
- **Seguridad:** La seguridad en la preservación digital implica proteger los datos almacenados contra accesos no autorizados o pérdidas de información. Se logra mediante medidas como encriptación, control de acceso y copias de seguridad. Esto cubre todo el ciclo de vida de los datos, desde el momento en el que se crean hasta el almace-

miento a largo plazo, es muy importante mantener la integridad y accesibilidad de la información a lo largo del tiempo [Gracy y Kahn, 2012].

2.3. Ejemplos de Preservación y Difusión del Legado Histórico en Plataformas Digitales

Es importante destacar que organizaciones que no se dedican a la preservación, como lo pueden ser los equipos de fútbol, también buscan estrategias para poder difundir su legado utilizando medios digitales. Dos casos que representan esto son el **FC Barcelona** y el **Real Madrid FC**, dos equipos ampliamente conocidos a nivel mundial por su trayectoria, estos logran contar su historia haciendo uso de sus páginas web de una forma atractiva y accesible.

2.3.1. FC Barcelona

El FC Barcelona, tiene un enfoque de utilizar tarjetas para representar su contenido. Estas tarjetas organizan la información de momentos claves del club, se conforman de imágenes llamativas y textos que cuenta logros del club, como se puede observar en Figura 3.



Figura 3: Web FC Barcelona.

Fuente: [FC Barcelona, 2024]

2.3.2. Real Madrid CF

Por otro lado, el Real Madrid CF, utiliza una línea de tiempo que ayuda a recorrer de forma cronológica sus hitos más importantes, lo cual ayuda a mostrar esta información de manera continua y permite al usuario tener una visión completa de la evolución del club a lo largo del tiempo, esto se puede ver de mejor forma en la Figura 4.

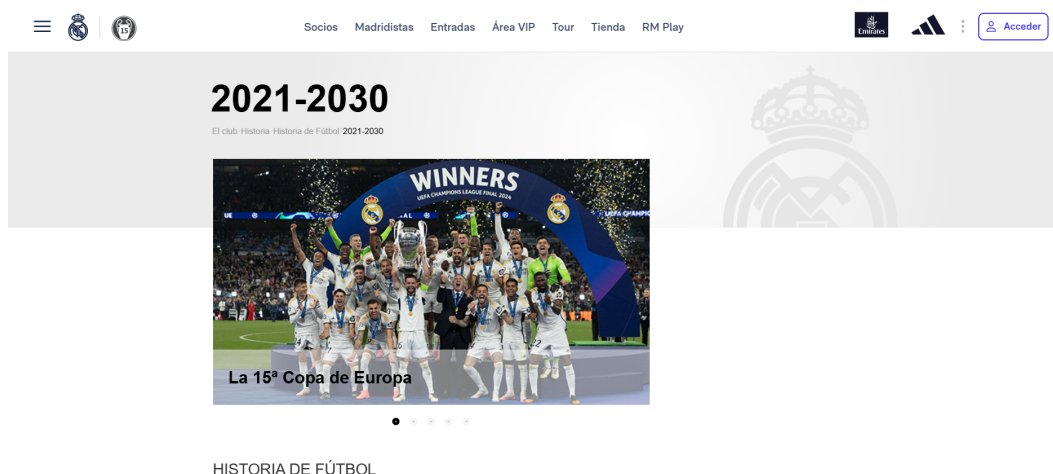


Figura 4: Web Real Madrid CF.

Fuente: [Real Madrid CF, 2024]

Aunque la preservación histórica no es el objetivo principal del fútbol, estos ejemplos demuestran cómo los clubes han utilizado sus plataformas digitales para preservar y difundir su legado, asegurando que su historia permanezca accesible.

2.4. Modelo OAIS

El Open Archival Information System (OAIS) es un modelo conceptual que tiene como propósito principal gestionar archivos y preservar documentos a largo plazo. Surgió de la Council of the Consultative Committee for Space Data Systems (CCSDS) a solicitud de la Organización Internacional de Normalización (ISO), con el objetivo de establecer un estándar claro y preciso para el almacenamiento de contenido digital generado por misiones espaciales [Lavoie, 2014].

Pese a que su origen estuvo enfocado en un dominio específico, el OAIS ha evolucionado y se ha adaptado para enfrentar desafíos similares en diversas organizaciones. A lo largo del tiempo, ha experimentado modificaciones que lo han convertido en un modelo versátil capaz de brindar apoyo en diferentes dominios, siendo aplicable en diversos contextos donde la preservación a largo plazo de información sea fundamental. Esta adaptabilidad y flexibilidad han permitido que el OAIS se convierta en una referencia valiosa y confiable para la gestión de archivos y documentos en múltiples ámbitos, proporcionando un modelo apropiado para garantizar la preservación y accesibilidad de datos digitales a lo largo del tiempo [Lavoie, 2014].

2.4.1. Roles en OAIS

Como se logra ver en Figura 5, hay 3 actores o roles fundamentales que interactúan externamente con el modelo OAIS, los cuales son:

- **Producer:** Es el rol o aplicación de suministrar la información que se desea preservar [CCSDS, 2019].
- **Consumer:** Es aquel rol desempeñado por aquellas personas o clientes que interactúan con los servicios de OAIS para conseguir información almacenada allí, es importante que los Consumer's deben ser capaces de entender la información que se les entrega, es decir existe un acuerdo entre ambas partes por lo que también podría tener el rol de Producer. [CCSDS, 2019].
- **Managment:** Es el rol que tienen aquellos que establecen las políticas generales del OAIS, es una función más de gestión que de administración interna del funcionamiento técnico, por lo que pueden verse involucradas organizaciones mas grandes [CCSDS, 2019].

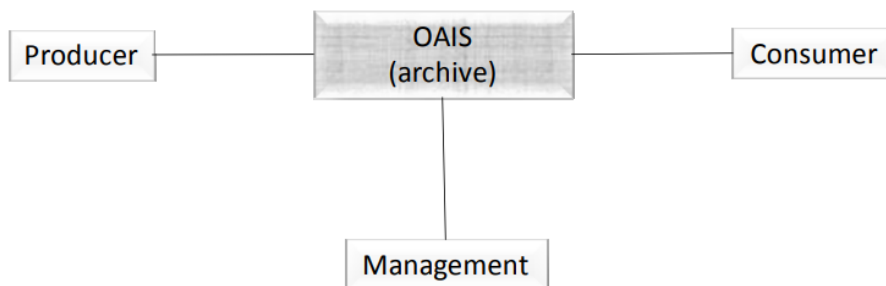


Figura 5: Entorno de OAIS.

Fuente: [CCSDS, 2019]

2.4.2. Tipos de Paquetes

El modelo OAIS también plantea un mecanismo para manejar los distintos archivos que pasan a través de él, por lo que propone el uso de un elemento general conocido como **Information Package (IP)**, esto lo que hace es generar un canal de transmisión discreto de elementos, sin tener una dependencia de los diferentes tipos que hay, sino que poder utilizar algo que los pueda encapsular.

Como se logra ver en la Figura 6, el IP se divide en dos tipos:

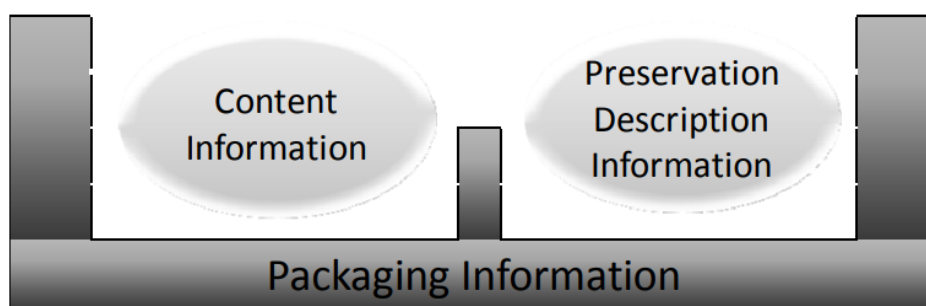


Figura 6: Information Package (IP).

Fuente: [CCSDS, 2019]

- **Content Information:** Es aquella información original que se desea preservar, es decir el **Data Object (DO)**, un concepto el cual tiene como finalidad poder generalizar cualquier tipo de información o archivo que se desee almacenar haciendo una simili-

tud con la orientación a objetos, por lo que es fundamental definir la estructura que tendrá el DO dentro del modelo OAIS [CCSDS, 2019].

- **Preservation Description Information (PDI):** Este elemento no puede existir sin antes tener bien definido el Content Information, luego de este punto el PDI es el encargado de gestionar y almacenar toda aquella información necesaria para la preservación [CCSDS, 2019], como lo son:

- Procedencia.
- Contexto.
- Referencia.
- Integridad.
- Derechos de acceso.

2.4.3. Tipos de Comunicación

Una vez definido como se utiliza la información dentro del modelo OAIS, es necesario mencionar el como se traspa la información entre los diferentes roles, es por ello que se plantea el tener variaciones de IPs que cumplan con los requisitos necesarios en su contexto específico, por lo que las distintas variantes son:

- **Submission Information Package (SIP):** Es el paquete que es enviado desde el Pro-ducer hacia el OAIS, su contenido y formato es acordado desde antes entre el Pro-ducer y el OAIS, la mayoría de estos paquetes contienen Content Information y PDIs [CCSDS, 2019].
- **Archival Information Packages (AIPs):** Este tipo de paquete engloba uno o varios SIPs que se transforman en uno o varios paquetes denominados AIPs (Archival Information Packages). Estos AIPs contienen una serie de PDI (Preservation Description Information) que se utilizan para establecer la forma en la que estos elementos serán pre-servados. Aquí es donde cobra gran relevancia la definición del OAIS, ya que en este punto los AIPs deben cumplir con los estándares internos establecidos por el OAIS [CCSDS, 2019].

- **Dissemination Information Package (DIP):** El OAIS en respuesta a una solicitud entrega parte de un AIP a un consumidor mediante un DIP (Dissemination Information Package), es aquí que según la petición que se haga es la estructura que tendrá el DIP, pero será acorde a la respuesta esperada por el Consumer [CCSDS, 2019].

En la Figura 7 podemos ver cómo es la interacción entre los diferentes roles del modelo OAIS, además de cómo estos se van comunicando mediante los distintos paquetes que se tienen a disposición.

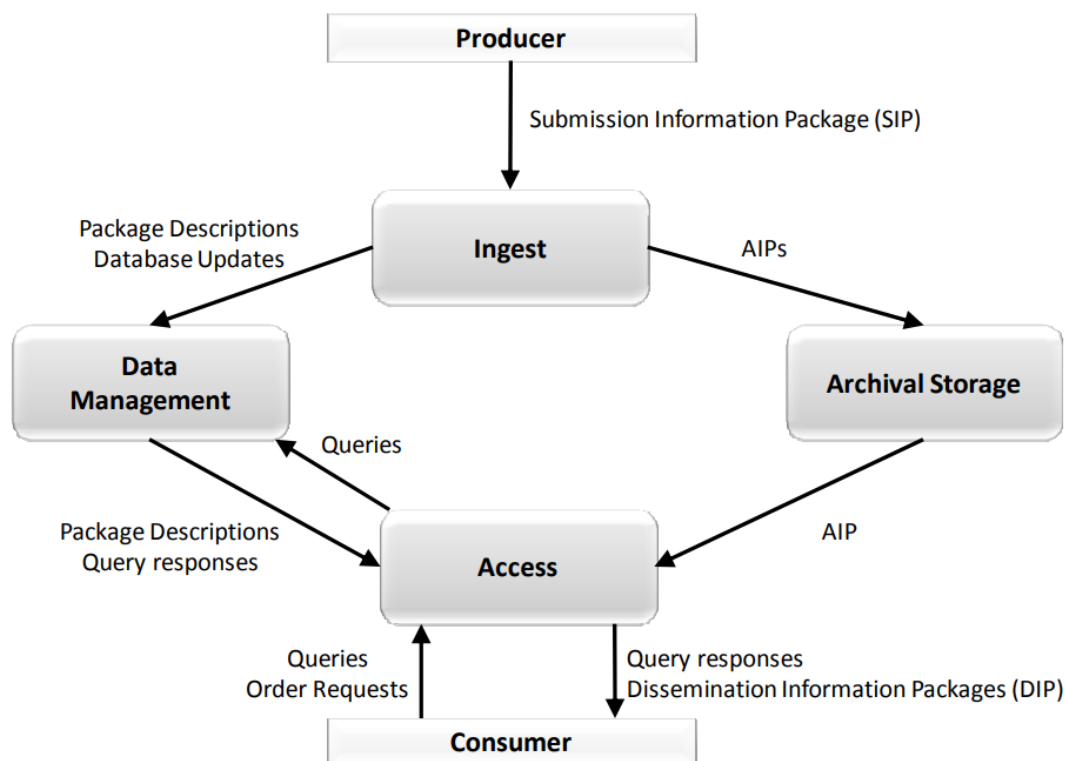


Figura 7: Diagrama de flujo de datos OAIS.

Fuente: [CCSDS, 2019]

2.5. Content Management System (CMS)

Un Content Management System (CMS), es una herramienta que permite crear, modificar y administrar contenidos en un sitio web sin la necesidad de tener conocimientos de programación, de esta forma se simplifica la gestión de los contenidos brindando una interfaz intuitiva

que permite a los usuarios editar y administrar información en tiempo real, sin la necesidad de escribir código [Ticona Condori, 2022].

Un buen CMS debe ofrecer ciertas características esenciales para asegurar su eficiencia y facilidad de uso:

- **Interfaz Intuitiva:** Debe contar con una interfaz clara, que permita el fácil entendimiento de cómo gestionar los contenidos que se tienen en el sistema.
- **Editor WYSIWYG:** Un editor **WYSIWYG (What You See Is What You Get)** o también conocidos como **Rich Text Editor** permiten a los usuarios utilizar texto e imágenes sin la necesidad de escribir en HTML, proporcionan una interfaz para crear contenidos de manera visual y sencilla.
- **Gestión de Usuarios:** Los CMS permiten la creación de variados tipos de usuarios con diferentes permisos y acceso a la información.
- **Extensibilidad:** La mayoría de CMS pueden agregar nuevas funcionalidades mediante la creación de módulos o plugins, permitiendo adaptar cada proyecto a una necesidad mas específica [Ticona Condori, 2022].

Como en todo sistema de software se puede hacer un análisis de sus ventajas y desventajas:

- **Ventajas:** Los CMS permiten gestionar contenidos de forma rápida y sencilla, tienen una capacidad de escalabilidad y expansibilidad muy buena por lo que son ideales para proyectos que requieren crecimiento continuo. Por otra parte su fácil implementación ayuda a reducir los tiempos y costos asociados, lo cual es atractivo para proyectos pequeños y medianos [Ticona Condori, 2022].
- **Desventajas:** Los CMS están limitados por sus módulos y plugins, esto provoca que si es necesario una funcionalidad específica no pueda ser gestionada por estos, además que la migración de un CMS a otro en la mayoría de los casos requiere un trabajo muy costoso. Además, al ser la mayoría de código abierto tienden a ser vulnerables a ataques si es que no se realizan actualizaciones periódicamente [Ticona Condori, 2022].

2.6. Front-end

2.6.1. Vue JS

Vue.js es un framework progresivo para construir interfaces de usuarios interactivas y dinámicas en aplicaciones web, entre sus ventajas está su **Reactividad**, permitiendo que los datos cambien en tiempo real sin la necesidad de recargar la página, también su sistema de **Componentes**, el cual permite dividir la interfaz en pequeñas piezas reutilizables donde cada componente es capaz de manejar su propia lógica interna, lo cual facilita el desarrollo y la mantenibilidad del código [Vue.js, 2024].

2.6.2. Vuetify

Vuetify es una biblioteca de componentes para Vue.js, la cual proporciona gran variedad de componentes predefinidos listos para usar en la aplicación, lo que ayuda a desarrollar de una forma más eficiente y poder adaptarlo a contextos específicos.

2.6.3. Axios

Axios es un Cliente HTTP, el cual permite realizar peticiones desde el cliente o node.js, esta biblioteca proporciona una forma simple y fácil para interactuar con servicios web o APIs.

2.7. Back-end

El Back-end es la parte en el desarrollo web que se encarga de manejar la lógica de negocio, el procesamiento de los datos y la comunicación con las bases de datos u otros sistemas relacionados.

Se ejecuta en un servidor, donde reside la funcionalidad principal de la aplicación. Los diferentes sistemas en conjunto proporcionan que los datos sean manejados de manera eficiente, segura y accesible para los usuarios finales que usarán la capa de **Front-end** para poder visualizar y gestionar la información.

2.7.1. API Rest

Una API Rest (Representational State Transfer), es un estilo de arquitectura para servicios web, utiliza el protocolo HTTP para la comunicación entre cliente y servidor [Amodeo, 2013].

Los sistemas basados en API Rest son escalables y permiten simplificar la integración con otros servicios, siguiendo una lógica para la construcción de micro servicios o sistemas modulares.

2.7.2. Express

Express es una infraestructura para aplicaciones web en Node.js, que proporciona un conjunto de características necesarias para tener unas bases sólidas para desarrollar proyectos web, como lo pueden ser aplicaciones web o API, entre sus características se encuentra:

- Manejo de rutas.
- Manejo de peticiones HTTP.
- Manejo de respuestas HTTP.

Una de las principales ventajas de Express es su fácil integración con otras bibliotecas o Frameworks, permitiendo el desarrollo personalizado de aplicaciones según la necesidad específica de cada proyecto [Express, 2024].

2.7.3. Object-Relational Mapping (ORM)

Es una técnica de programación que permite una fácil interacción entre las aplicaciones y la base de datos. Un ORM permite a los desarrolladores trabajar con los datos como si fuesen objetos del lenguaje de programación que están utilizando, de esta forma se evita el utilizar consultas SQL manuales, no solo simplifica el acceso a la base de datos sino que también mejora la mantenibilidad del código y evita posibles errores.

2.8. Object Storage

Es un tipo de arquitectura de almacenamiento de datos donde cada dato es tratado como un objeto individual, en lugar de como un simple archivo en sistemas de archivos tradicionales. Estos objetos contienen datos, metadata y un identificador único, lo cual proporciona una interfaz similar a la de un archivo, pero ofrece una abstracción al cómo se trabajan con estos [Factor *et al.*, 2005].

Este tipo de arquitectura permite una gran escalabilidad, ya que los objetos pueden distribuirse en servidores en la nube o de manera local sin la necesidad de reestructurar el sistema.

CAPÍTULO 3

PROPUESTA DE SOLUCIÓN

3.1. Levantamiento de Requerimientos

3.1.1. Alcance del Proyecto

Este proyecto tiene como objetivo preservar y difundir el legado de la Feria de Software, para ello se desarrolla una aplicación web que permite gestionar y visualizar una variedad de contenidos generados a lo largo de los años.

Es una aplicación web que no solo tiene la capacidad de almacenar y gestionar los diferentes contenidos multimedia que se ingresen, sino que también propone una forma de difusión para que todos estos contenidos puedan ser accesibles para cualquier persona que lo desee, beneficiando no solo al DI, sino que a la comunidad universitaria en general, al centralizar la información y brindar un mecanismo para su rápido acceso.

El alcance de la aplicación contempla las siguientes funcionalidades:

- **Visualización de Contenidos:** Permite a los usuarios navegar por diferentes contenidos y ediciones de la Feria de Software, accediendo a fotos, vídeos, documentos y publicaciones asociadas a cada evento.
- **Gestión de Contenidos:** La aplicación tiene un panel de administración donde se puede visualizar, editar, crear y eliminar los distintos contenidos disponibles.
- **Acceso a la Información:** Dispone de un mecanismo para acceder a la información desde diferentes sistemas, permitiendo que la información sea pública y de fácil acceso.
- **Escalabilidad:** Tiene un diseño escalable, con el objetivo de poder expandirse fácilmente en el futuro, facilitando agregar nuevas funciones u otros tipos de contenidos.

3.1.2. Marco del Proyecto

Se establecen las fases que se llevan a cabo durante el desarrollo del proyecto. Estas son:

- **Desarrollo del Frontend:** Fase en la cual se desarrolla la interfaz que utilizarán los usuarios para interactuar con la aplicación. Se da una atención especial a que el contenido pueda ser visualizado en los diferentes dispositivos en los que los usuarios consumen contenidos de aplicaciones web.
- **Desarrollo del Backend:** El backend se encarga de toda la gestión de los contenidos de la aplicación, tanto los contenidos multimedia como las interacciones de los usuarios con la información almacenada en base de datos, además de poder gestionar de buena forma el almacenamiento de los contenidos multimedia.
- **Desarrollo del Frontend - Administración:** Una vez desarrollado el backend se realiza la interfaz para que los usuarios administradores puedan interactuar y gestionar la información, es decir, crear, editar y eliminar nuevos contenidos.
- **Pruebas y Validación:** Una vez el desarrollo de la aplicación sea completado se deben realizar pruebas y validaciones con usuarios, probando sus funcionalidades y flujos críticos para comprobar que el comportamiento sea el esperado. Estas pruebas permiten identificar posibles fallos y mejoras a la experiencia de los usuarios, proporcionando información importante para hacer las adaptaciones necesarias a la aplicación.

3.1.3. Tipos de Contenidos

El sistema tiene la capacidad de manejar diferentes tipos de contenidos asociados a la Feria de Software, para ello es importante entender cuáles son los contenidos claves y que se han generado con el tiempo. Los contenidos que forman parte de la aplicación son:

- **Proyectos Destacados:** Son aquellos proyectos que generaron un impacto en la Feria de Software, como pueden ser aquellos que ganaron premios o impulsaron nuevas iniciativas.

- **Testimonios:** Este tipo de contenido es para dar a conocer la experiencia de equipos, estudiantes o personas destacadas que han participado en las distintas ediciones de la Feria de Software.
- **Investigaciones:** Hace referencia a todo lo relacionado con temas de investigación que nacen en la Feria de Software, como lo pueden ser papers científicos, memorias de alumnos, entre otros.
- **Impacto en los Medios:** Son contenidos que los medios de prensa han generado a raíz de la Feria de Software, como pueden ser notas en periódicos, noticias, páginas web, etc.
- **Hitos Históricos:** Son aquellos acontecimientos o hitos que marcaron a lo largo de los años la Feria de Software.
- **Galerías:** Conjunto de imágenes, vídeos y contenidos multimedia relacionados a la Feria de Software.

Dado el origen estructurado de los contenidos anteriores, podrían quedar fuera algunos elementos que no encajan dentro de esas categorías, es por ello que se incorpora un contenido adicional, al cual se le proporciona un nivel de abstracción que permite adaptarse a cualquier caso que no cumpla con la estructura anterior y que se le denomina **Preservaciones**.

- **Preservaciones:** Son todos aquellos elementos que no caben dentro de las categorías de los contenidos estándar de la aplicación, pero que aun así son fundamentales para preservar la historia de la Feria de Software, pueden ser representaciones de productos físicos o tangibles, los cuales tienen un valor simbólico y representan logros dentro de la Feria de Software, ejemplos de preservaciones podrían ser:
 - **Llaveros, insignias, merchandising:** Elementos que los equipos utilizan para dar a conocer sus productos y representan la identidad de cada una de las Ferias de Software.
 - **Prototipos Físicos:** Puede darse el caso que en la Feria de Software se generen productos tangibles que forman parte del proyecto.

- **Materiales Promocionales:** Folletos, carteles y cualquier otro elemento que pueda ser parte de la Feria de Software y los distintos equipos que participan en cada edición.

Estos elementos no siguen una estructura definida como la pueden tener los otros contenidos, pero son fundamentales de igual forma para una preservación completa de la Feria de Software. En el punto 3.3 se plantea con más detalle el mecanismo para gestionar estos contenidos.

3.2. Diseño Modelo de Datos

Dentro del modelo de datos se identifican 3 entidades principales que engloban toda la aplicación:

- **Feria**
- **Publicacion**
- **Preservacion**

Cada una de ellas cuenta con una tabla específica para sus archivos multimedia adjuntos. Este diseño se plantea para generar una mayor especialización de información relacionada a los contenidos multimedia que se asocian a cada una de estas entidades, permitiendo una mayor flexibilidad.

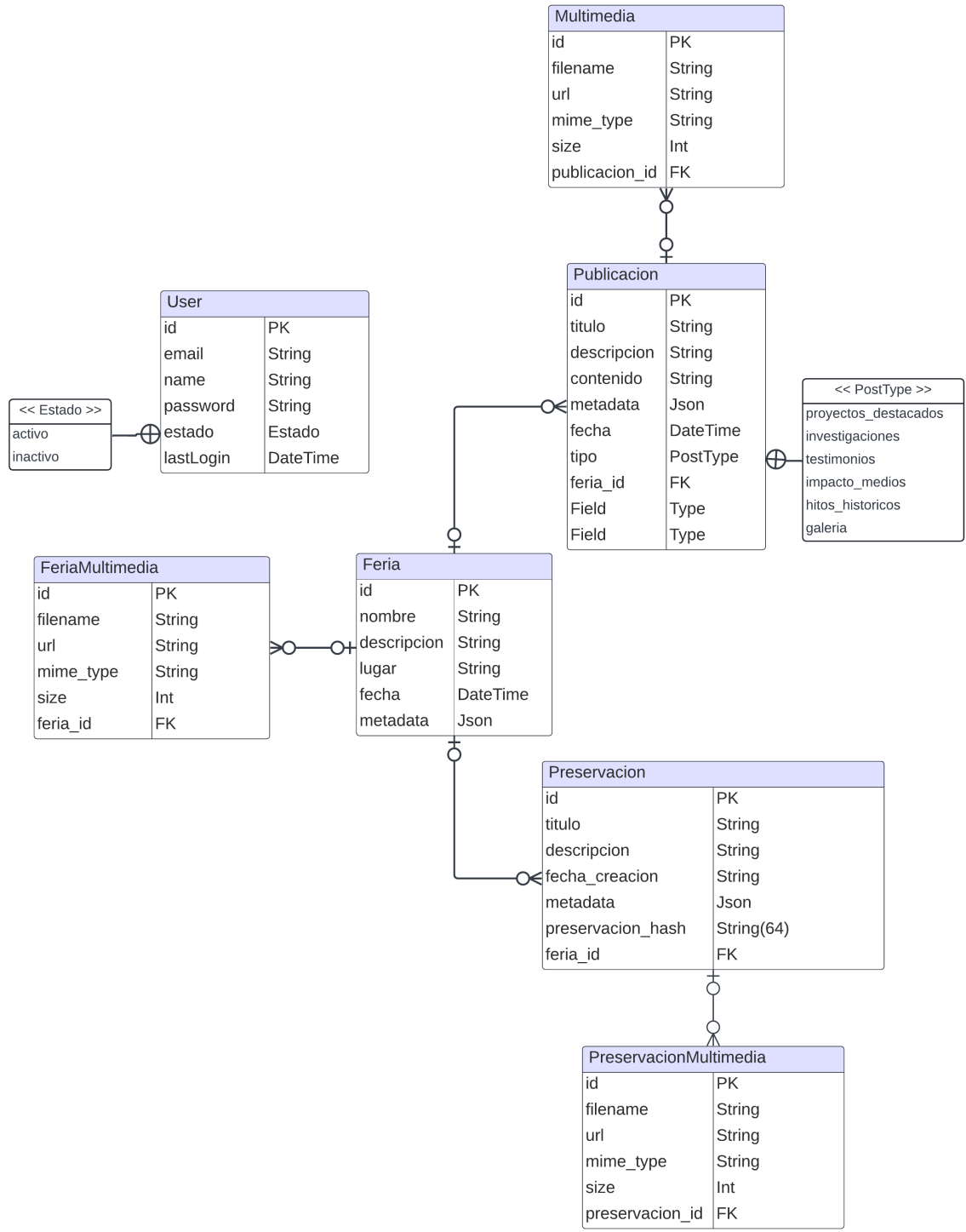


Figura 8: Modelo de Datos

Fuente: Elaboración propia

3.3. Diseño Paquete de Información

Teniendo en cuenta la gran variedad de información que se maneja en la aplicación, surge la necesidad de crear un paquete de información (IP) que asegure su preservación a lo largo del tiempo. En este sentido, el modelo OAIS proporciona una sólida referencia sobre cómo abordar esta tarea. Este paquete de información se compone de dos partes fundamentales: la generalización del almacenamiento de cualquier elemento y la asociación de diversos archivos que representan la información a preservar, como imágenes, vídeos, documentos, entre otros.

Por otro lado, es crucial poder vincular la información necesaria para la preservación. Dentro del proyecto, nos referimos a esto como "*metadata de preservación*". Dado que cada elemento a preservar es único en su naturaleza en especial cuando hablamos del tipo de contenido **Preservaciones**, establecer una estructura rígida para esta *metadata* no es algo que se deba hacer. Por eso, se da total libertad a quien crea los paquetes de información para que pueda incluir los elementos que mejor se ajusten al contexto de cada caso. Esta flexibilidad permite aprovechar la información disponible, evitando guardar datos innecesarios o incorrectos, y asegurando que cada elemento se adapte bien a su situación específica.

La **metadata de preservación** debe incluir información referente al elemento a preservar, como pueden ser, color, tipo de material, tamaño, origen, etc. Es aquí donde el entendimiento de cada elemento a preservar es fundamental para poder plasmarlo de la mejor manera posible con información clara y detallada para su correcta preservación a lo largo de los años.

3.4. Diseño API

En este proyecto, se decide utilizar una arquitectura de múltiples capas para organizar mejor la API. Se da un enfoque en dividir las diferentes partes de la API y aplicar buenas prácticas de desarrollo. Al separar y modularizar las distintas funciones del código, no solo se busca hacer el sistema más escalable, sino también facilitar su mantenimiento y evolución en el futuro.

3.4.1. Arquitectura

La API se estructura en tres capas principales, cada una de ellas con responsabilidades específicas:

- **Routes (Rutas):** Esta capa es responsable de definir las rutas o endpoints de la API. Las rutas determinan cómo los clientes pueden interactuar y qué acciones pueden realizar. Cada ruta define una URL y especifica qué métodos HTTP son permitidos (GET, POST, PUT, DELETE, etc.). En esta capa, se configuran los manejadores de rutas de Express para responder a las solicitudes HTTP entrantes y dirigirlos al controlador correspondiente, además de proveer una capa donde se puede verificar.
- **Controllers (Controladores):** Los controladores son responsables de manejar las solicitudes HTTP entrantes y realizar la lógica necesaria para esas solicitudes, estos interactúan con los servicios para obtener o manipular datos. Son el punto de entrada de las solicitudes HTTP y están diseñados para mantenerse lo más abstractos posibles, delegando la mayor parte de la lógica de negocio a los servicios.
- **Services (Servicios):** La capa de servicios contiene la lógica de negocio. Los servicios son responsables de interactuar con la base de datos para realizar operaciones CRUD (Create, Read, Update, Delete) y cualquier otra operación relacionada con la lógica de negocio. Aquí es donde se realizan validaciones de datos y se ejecutan operaciones complejas que no pertenecen a la capa de controladores, como lo son la subida de archivos, ordenar las consultas a la base de datos, etc.

En la Figura 9 se puede ver de mejor manera el flujo y las capas mencionadas anteriormente:

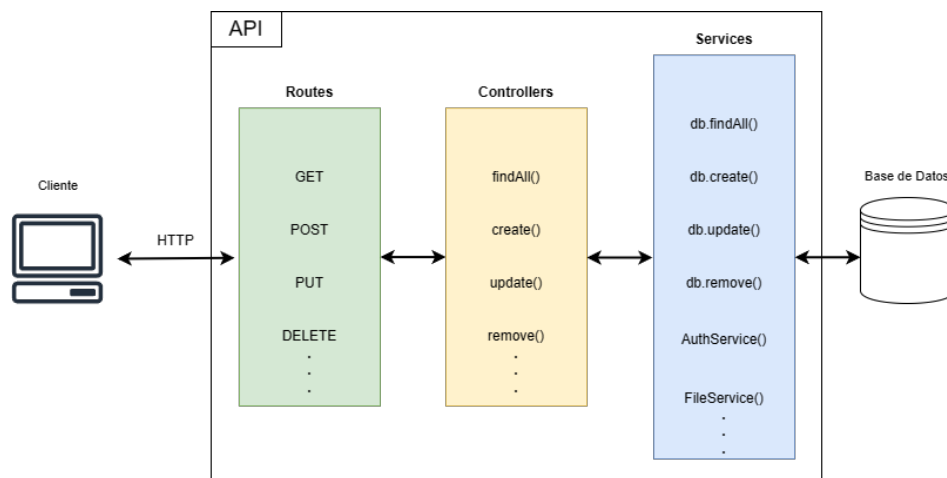


Figura 9: Capas internas API

Fuente: Elaboración propia

Además se cuenta con **Middlewares** los cuales se encargan de manejar ciertas tareas antes de que las peticiones lleguen a los controladores. Son útiles para asegurar que todo funcione de manera correcta, en el contexto del proyecto se cuenta con dos Middleware sumamente importantes:

- **FileUploadMiddleware:** Este Middleware se encarga de procesar las solicitudes HTTP que contienen archivos en el formato **form-data**, toma estos archivos y los deja de una manera más simple para que la aplicación pueda utilizarlos más fácilmente. Son utilizados en el contexto de solicitudes donde se Crean o Modifican datos, es decir, con los métodos HTTP POST y PATCH.
- **AuthMiddleware:** Este middleware se encarga de validar que el **auth-token** enviado en la cabecera de la solicitud sea válido, es decir que haya sido generado por un usuario registrado y activo en la aplicación. Se encarga de la autenticación y es especialmente importante, ya que es usado para aquellos métodos HTTP que alteran datos, es decir, Crear, Editar o Eliminar. Así se garantiza que solo usuarios autorizados puedan realizar estas peticiones que alteran la información. Se dará mayor detalle en el siguiente punto 3.4.2.

3.4.2. Seguridad

Se plantea un mecanismo de seguridad para poder controlar el acceso y la manipulación de los datos. Este mecanismo de autenticación y autorización está basado en **JSON Web Token (JWT)** y permite que únicamente los usuarios que se tienen registrados en la base de datos puedan realizar operaciones sensibles, como lo son crear, modificar y eliminar. El proceso para la validación es el siguientes:

Generación y uso de JWT:

- **Autenticación mediante JWT:** Al momento de iniciar sesión un usuario se genera el token JWT, se utilizará en todas las solicitudes que realice el usuario en la API.
- **Duración JWT:** El token tiene una vigencia de **24 horas**, esto también se considera como un mecanismo de seguridad ya que si el token es robado solo sirve en la duración de su vida útil; como medida adicional frente a un ataque es posible darle el estado de "inactivo" al usuario, lo cual hará que cualquier token vigente generado por este sea rechazado por la API.
- **Transporte JWT:** El token debe ser enviado en la cabecera HTTP con el nombre **auth-token**, este se debe enviar en cada solicitud que requiera autenticación, la API se encarga de validar esto en cada petición, en caso de que el token no esté siendo enviado la petición es rechazada y devuelve el mensaje **"No token provided"**.

Validación del JWT:

El proceso de validación del Token es mediante **AuthMiddleware**, este se encarga de validar el token recibido en la cabecera HTTP, asegurando que sea valido, que no esté expirado y que pertenezca a un usuario creado y activo en base de datos.

- **Validar token:** Se verifica que el token no se haya alterado o manipulado.
- **Usuario activo:** Se valida que el usuario asociado al token tenga el estado "activo" en base de datos, en caso contrario la petición es rechazada y devuelve el mensaje **"Unauthorized"**.

- **Métodos protegidos:** Los métodos protegidos HTTP son **POST**, **PATCH** y **DELETE**, todos estos métodos están relacionados a una modificación de datos, de esta forma se garantiza que solo usuarios con permisos puedan realizar estas operaciones.

Proceso de Autorización:

Para asegurar que solo usuarios autorizados puedan realizar operaciones de modificación de datos, **AuthMiddleware** se aplica a cada uno de los endpoints de la API donde se tengan que aplicar métodos críticos, tales como:

- Creación de nuevos registros.
- Actualización de información existente.
- Eliminar un registro en base de datos.

Teniendo este enfoque se da protección a todas aquellas acciones críticas y se previene el acceso no autorizado que pueda llegar a comprometer la seguridad de la API y la información que se está almacenando.

3.4.3. Endpoints

- **Endpoints de Autenticación:** Estos son los endpoints encargados del manejo de usuarios y validación de tokens.

post	/api/v1/login/ <i>Recibe las credenciales de un usuario y devuelve un token valido (No necesita autenticación).</i>
post	/api/v1/register/ <i>Se encarga de registrar un nuevo usuario, solo puede usarlo un usuario registrado (Necesita autenticación).</i>

post	/api/v1/verify/ <i>Valida si el token enviado en la cabecera HTTP auth-token es valido (No necesita autenticación).</i>
get	/api/v1/users/ <i>Obtiene los usuarios (Necesita autenticación).</i>
get	/api/v1/users/:id <i>Obtiene un usuario por su ID (Necesita autenticación).</i>
patch	/api/v1/users/:id <i>Actualiza un usuario por su ID (Necesita autenticación).</i>
delete	/api/v1/users/:id <i>Elimina un usuario por su ID (Necesita autenticación).</i>

- **Endpoints de Publicaciones:** Este es uno de los endpoints más importantes, ya que en este se manejan los diferentes tipos de contenido que se podrán gestionar en la API, a continuación una lista de las publicaciones disponibles:

■ **Proyectos Destacados:**

```
{  
  titulo: "Esto es un titulo.",  
  descripcion: "Esto es una descripción.",  
  contenido: "Esto es el contenido.",  
  tipo: "proyectos_destacados",  
  feria_id: 1,  
  multimedia: [.....],  
}
```

get	/api/v1/publicaciones/proyectos-destacados <i>Obtiene las publicaciones de tipo proyectos destacados.</i>
post	/api/v1/publicaciones/proyectos-destacados <i>Crea una publicación de tipo proyectos destacados (Necesita autenticación).</i>
get	/api/v1/publicaciones/proyectos-destacados/:id <i>Obtiene una publicación de tipo proyectos destacados por su ID.</i>
patch	/api/v1/publicaciones/proyectos-destacados/:id <i>Actualiza una publicación de tipo proyectos destacados por su ID (Necesita autenticación).</i>
delete	/api/v1/publicaciones/proyectos-destacados/:id <i>Elimina una publicación de tipo proyectos destacados por su ID (Necesita autenticación).</i>

■ **Impacto en los Medios:**

```
{
  titulo: "Esto es un titulo.",
  descripcion: "Esto es una descripción.",
  contenido: "Esto es el contenido.",
  fecha: "2024-08-22 04:00:00",
  tipo: "impacto_medios",
  feria_id: 1,
  metadata: {
    autor: "Nombre Autor",
  },
  multimedia: [.....],
}
```

get	/api/v1/publicaciones/impacto-medios <i>Obtiene las publicaciones de tipo impacto en los medios.</i>
post	/api/v1/publicaciones/impacto-medios <i>Crea una publicación de tipo impacto en los medios (Necesita autenticación).</i>
get	/api/v1/publicaciones/impacto-medios/:id <i>Obtiene una publicación de tipo impacto en los medios por su ID.</i>
patch	/api/v1/publicaciones/impacto-medios/:id <i>Actualiza una publicación de tipo impacto en los medios por su ID (Necesita autenticación).</i>
delete	/api/v1/publicaciones/impacto-medios/:id <i>Elimina una publicación de tipo impacto en los medios por su ID (Necesita autenticación).</i>

■ Investigaciones:

```
{
  titulo: "Esto es un titulo.",
  descripcion: "Esto es una descripción.",
  contenido: "Esto es el contenido.",
  fecha: "2024-08-22 04:00:00",
  tipo: "investigaciones",
  feria_id: 1,
  metadata: {
    autor: "Nombre Autor",
  },
  multimedia: [.....],
}
```

get	/api/v1/publicaciones/investigaciones <i>Obtiene las publicaciones de tipo investigaciones.</i>
post	/api/v1/publicaciones/investigaciones <i>Crea una publicación de tipo investigaciones (Necesita autenticación).</i>
get	/api/v1/publicaciones/investigaciones/:id <i>Obtiene una publicación de tipo investigaciones por su ID.</i>
patch	/api/v1/publicaciones/investigaciones/:id <i>Actualiza una publicación de tipo investigaciones por su ID (Necesita autenticación).</i>
delete	/api/v1/publicaciones/investigaciones/:id <i>Elimina una publicación de tipo investigaciones por su ID (Necesita autenticación).</i>

■ Testimonios:

```
{  
  titulo: "Esto es un titulo.",  
  descripcion: "Esto es una descripción.",  
  tipo: "testimonios",  
  feria_id: 1,  
  metadata: {  
    puntaje: "5",  
  },  
  multimedia: [.....],  
}
```

get

/api/v1/publicaciones/testimonios

Obtiene las publicaciones de tipo testimonios.

post

/api/v1/publicaciones/testimonios

Crea una publicación de tipo testimonios (Necesita autenticación).

get

/api/v1/publicaciones/testimonios/:id

Obtiene una publicación de tipo testimonios por su ID.

patch

/api/v1/publicaciones/testimonios/:id

Actualiza una publicación de tipo testimonios por su ID (Necesita autenticación).

delete

/api/v1/publicaciones/testimonios/:id

Elimina una publicación de tipo testimonios por su ID (Necesita autenticación).

■ **Galería:**

```
{  
  titulo: "Esto es un titulo.",  
  descripcion: "Esto es una descripción.",  
  fecha: "2024-08-22 04:00:00",  
  tipo: "galeria",  
  feria_id: 1,  
  multimedia: [.....],  
}
```

get

/api/v1/publicaciones/galeria

Obtiene las publicaciones de tipo galería.

post /api/v1/publicaciones/galeria

Crea una publicación de tipo galería (Necesita autenticación).

get /api/v1/publicaciones/galeria/:id

Obtiene una publicación de tipo galería por su ID.

patch /api/v1/publicaciones/galeria/:id

Actualiza una publicación de tipo galería por su ID (Necesita autenticación).

delete /api/v1/publicaciones/galeria/:id

Elimina una publicación de tipo galería por su ID (Necesita autenticación).

■ Hitos Históricos:

```
{
  titulo: "Esto es un titulo.",
  descripcion: "Esto es una descripción.",
  fecha: "2024-08-22 04:00:00",
  tipo: "hitos_historicos",
  multimedia: [.....],
}
```

get /api/v1/publicaciones/hitos-historicos

Obtiene las publicaciones de tipo hitos históricos.

post /api/v1/publicaciones/hitos-historicos

Crea una publicación de tipo hitos históricos (Necesita autenticación).

get

/api/v1/publicaciones/hitos-historicos/:id

Obtiene una publicación de tipo hitos históricos por su ID.

patch

/api/v1/publicaciones/hitos-historicos/:id

Actualiza una publicación de tipo hitos históricos por su ID (Necesita autenticación).

delete

/api/v1/publicaciones/hitos-historicos/:id

Elimina una publicación de tipo hitos históricos por su ID (Necesita autenticación).

- Endpoints de Archivos:

post

/api/v1/files/feria

*Crea un nuevo archivo en la entidad feria dado el query param **relation_id** (Necesita autenticación).*

delete

/api/v1/files/feria/:id

Elimina un archivo con una ID específica, asociado a una feria (Necesita autenticación).

post

/api/v1/files/publicacion

*Crea un nuevo archivo en la entidad publicación dado el query param **relation_id** (Necesita autenticación).*

delete

/api/v1/files/publicacion/:id

Elimina un archivo con una ID específica, asociado a una publicación (Necesita autenticación).

post

/api/v1/files/preservacion

*Crea un nuevo archivo en la entidad preservación dado el query param **relation_id** y **preservacion_hash** (Necesita autenticación).*

delete

/api/v1/files/preservacion/:id

Elimina un archivo con una ID específica, asociado a una preservación (Necesita autenticación).

- Endpoints de Preservación:

```
{
  titulo: "Esto es un titulo.",
  descripcion: "Esto es una descripción.",
  fecha_creacion: "2024",
  feria_id: 1,
  preservacion_hash: "7c5dbaa802289b81c1a709...",
  metadata: { .... },
  preservacion_multimedia: [.....],
}
```

get

/api/v1/preservacion

Obtiene las preservaciones.

post

/api/v1/preservacion

Crea una nueva preservación (Necesita autenticación).

get

/api/v1/preservacion/:id

Obtiene una preservación por su ID.

patch **/api/v1/preservacion/:id**

Actualiza una preservación por su ID (Necesita autenticación).

delete **/api/v1/preservacion/:id**

Elimina una preservación por su ID (Necesita autenticación).

3.5. Almacenamiento de Datos y Archivos

Tener una correcta gestión y almacenamiento de los datos y archivos multimedia es un pilar fundamental en el desarrollo de esta aplicación. Debido a la diversidad y cantidad de contenidos que se tienen que almacenar, se propone una solución de almacenamiento que utiliza una **base de datos relacional** para gestionar todo tipo de datos estructurados, y por otro lado, el uso de un **sistema de almacenamiento de objetos (Object Storage)** en el cual se almacenaran todos los archivos multimedia.

3.5.1. Base de Datos PostgreSQL

PostgreSQL se utiliza como base de datos principal del sistema, esto debido a su gran compatibilidad y soporte con cualquier tipo de plataforma o arquitectura, en el caso del proyecto, PostgreSQL se encargara de almacenar:

- **Contenidos:** Toda la información de los contenidos se encontrará almacenado en las tablas de la base de datos, como lo son, título, descripción, fecha, tipo de contenido y enlaces a los archivos almacenados en MinIO 3.5.2.
- **Usuarios:** Datos que estén relacionados a los usuarios disponibles en la aplicación, nombre, email, estado, contraseña encriptada, último inicio de sesión, etc.
- **Ferias de Software y relaciones a los contenidos:** PostgreSQL se encargará de manejar todas las relaciones que existen entre los contenidos y las Ferias de Software, de tal

forma que se pueda consultar la información acorde a la necesidad específica que se tenga y poder realizar filtrados más eficientes.

- **Información Contenidos Multimedia:** Toda la información básica relacionada a los archivos almacenados en MinIO se guarda en esta base de datos para poder facilitar su gestión y la relación que tiene con cada tipo de contenido, la información que se almacena son por ejemplo, nombre del archivo, peso del archivo, enlaces al archivo y las relaciones que tiene con los distintos contenidos.

3.5.2. MinIO Object Store

Para poder almacenar la gran cantidad de contenidos multimedia como videos, fotos y otros contenidos a preservar, se utiliza **MinIO**, el cual es un sistema de almacenamiento de objetos, es un sistema distribuido lo que permite manejar grandes volúmenes de datos no estructurados.

En el contexto del proyecto, MinIO se encargará de almacenar:

- **Archivos Multimedia:** Todos los archivos multimedia estarán alojados y gestionados por MinIO.
- **Acceso eficiente y seguro:** Permite realizar operaciones rápidas de lectura y escritura de archivos sin grandes tiempos de espera, además está configurado para tener un acceso restringido y solo se tienen permisos de lectura para que puedan obtener los archivos almacenados, de esta forma se previene que puedan ser modificados por usuarios no deseados.

3.6. Diseño Interfaz

Para el diseño de la interfaz de la aplicación web se plantea un diseño modular, utilizando componentes, generando una estructura clara y escalable, en este caso se utiliza **Vuetify**, una biblioteca de componentes del Framework **Vue.js**, de esta forma se tiene una base para construir elementos adaptables a cualquier tipo de dispositivos, además de facilita la cons-

trucción de nuevos componentes específicos y abordar de mejor forma el diseño de una interfaz.

3.6.1. Componentes Transversales

Estos son componentes ampliamente utilizados en la aplicación, por lo que tener una explicación más clara de cada uno es fundamental para comprender su funcionamiento y cómo estos son capaces de integrarse en muchas partes, además permiten mantener una coherencia y estandarización en la aplicación.

- **Menú de Navegación:** El menú de navegación es una parte fundamental dentro de la aplicación, permite a los usuarios dirigirse a los diferentes contenidos disponibles, además se le da gran importancia a ofrecer una experiencia óptima, es por ello que el menú de navegación está diseñada de manera adaptable para dispositivos móviles, como se puede ver en Figura 11.

El menú de navegación se ubica en la parte superior de la aplicación, se encuentra fijo allí para que esté siempre visible para los usuarios, de tal forma que puedan acceder en cualquier momento.

Los elementos que tiene el menú de navegación son, de izquierda a derecha, primero se encuentra el logo de la Feria de Software, el cual tiene la característica que al ser pulsado vuelve a la página inicial de la aplicación, los demás elementos son las diferentes secciones dentro de la aplicación, las cuales están ordenadas de la siguiente manera:

- Inicio
- Historia
- Preservaciones
- Galerías
- Otros
 - Proyectos Destacados
 - Testimonios

- Investigaciones
- Impacto en los Medios

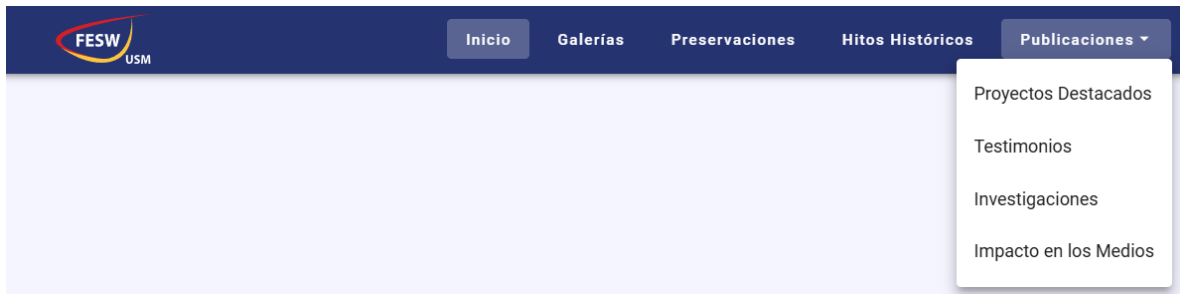


Figura 10: Menú de Navegación

Fuente: Elaboración propia

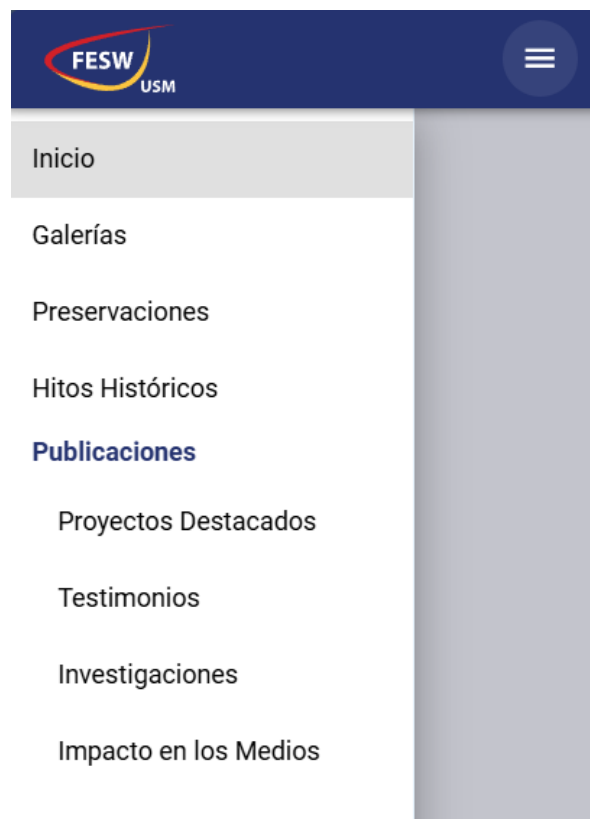


Figura 11: Menú de Navegación versión mobile

Fuente: Elaboración propia

- **Selector Feria de Software Global:** Este componente es uno de los más importantes, al igual que el menú de navegación. Permite seleccionar una de las Ferias de Software registradas en la aplicación. Una vez que se ha realizado una selección, toda la aplicación muestra únicamente contenidos relacionados con esa Feria de Software específica. Este comportamiento resulta ideal cuando se maneja una gran cantidad de información y se desea enfocar únicamente en los datos asociados a una feria en particular.

Se encuentra ubicado en la esquina superior derecha justo abajo de menú de navegación, para dispositivos móviles estará ubicado en el mismo lugar solo que de manera centrada.

Al momento de hacer clic en el botón **"SELECCIONA UNA FERIA DE SOFTWARE"** se despliega en medio de la pantalla una pequeña ventana para seleccionar la Feria de Software deseada, como se puede observar en Figura 12, una vez seleccionada, el botón muestra el nombre de la Feria de Software además de su fecha, dando la posibilidad que la navegación dentro de la aplicación solo muestre contenidos de esa Feria de Software.

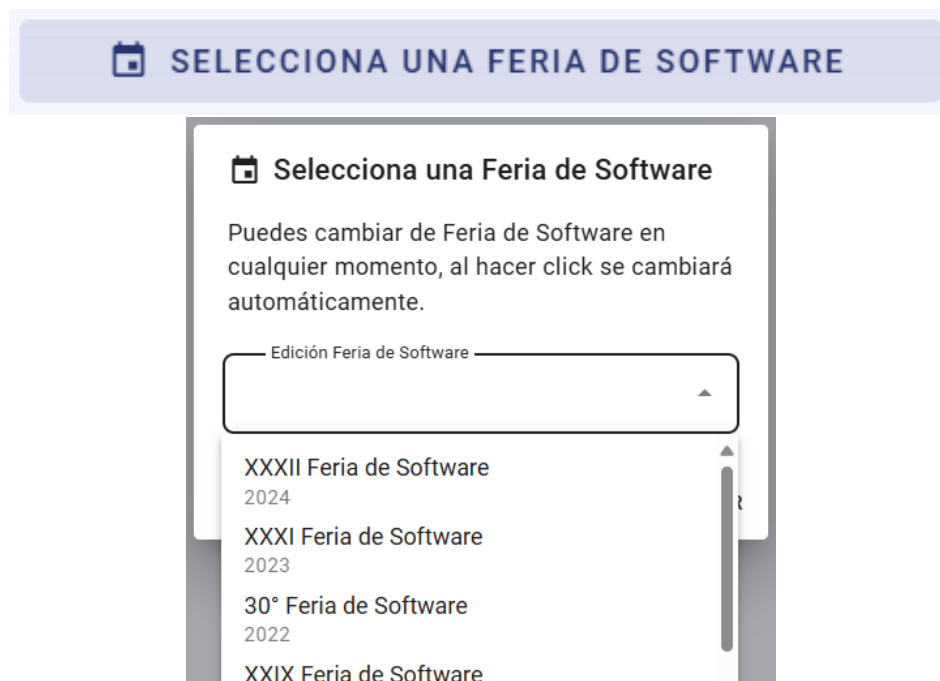


Figura 12: Componente FeriaSelectGlobal

Fuente: Elaboración propia

- **Opciones Administrador:** Este componente es exclusivo para usuarios administradores. Una vez que el usuario administrador inicia sesión, el componente aparecerá en la esquina inferior derecha de la interfaz. Se mantendrá fijo en toda la aplicación, permitiendo al administrador acceder a él en cualquier momento.

El componente ofrece tres opciones clave:

- **Logout:** cierra la sesión del usuario administrador.
- **Home:** Lleva al usuario administrador a las Vistas Generales de la aplicación.
- **Vista Admin:** Lleva al usuario administrador a la vista de administración.

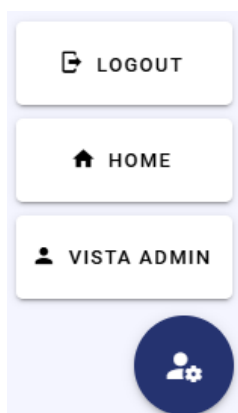


Figura 13: Componente AdminOptions

Fuente: Elaboración propia

- **Subida y gestión de archivos:** Este componente es de la parte administrativa de la aplicación, está presente en los formularios de creación y de edición de contenidos que tengan asociados el manejo de archivos, cada archivo es representado como una pequeña tarjeta, esta se compone de una imagen o icono que lo represente, el nombre del archivo, el tamaño del archivo y un botón para poder borrarlo, esto se puede ver de mejor forma en la Figura 14.

El componente se divide en 3 partes principales:

- **Visualización de archivos:** Muestra los archivos actualmente almacenados, desde esta sección el usuario administrador podrá descargar o eliminar los archivos.
- **Área de carga de archivos:** Permite explorar el dispositivo del usuario para poder seleccionar los diferentes archivos que desee subir a la aplicación, además para una mejor experiencia se implementa la capacidad de arrastrar archivos.

- **Vista previa archivos:** Muestra una vista previa de los archivos seleccionados antes de subirlos, facilitando la revisión de los contenidos.

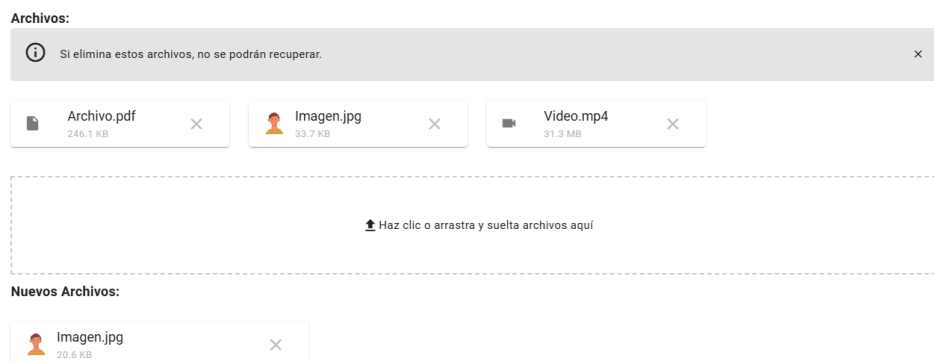


Figura 14: Componente FileUploader

Fuente: Elaboración propia

- **Componente Galería:** Componente encargado de mostrar los diferentes archivos asociados a los distintos tipos de recursos en la aplicación, es ampliamente utilizado sobre todo en las vistas individuales de cada publicación, por otra parte el componente está integrado con una biblioteca de JavaScript que permite la visualización de estos contenidos en pantalla completa haciendo uso de un lightbox; la biblioteca es *lightGallery* [LightGallery, 2024].

Este componente consta de de dos partes:

- **Archivos adjuntos:** Un botón que despliega una lista de archivos que no tienen una estructura definida y no son compatibles con la visualización en un entorno web nativo. Al hacer clic en cualquiera de estos archivos, se inicia su descarga automáticamente.
- **Cuadrícula de archivos:** Esta sección muestra todos los archivos o contenidos multimedia que pueden ser visualizados directamente en la interfaz web. Al hacer clic en uno de estos archivos, se abre en pantalla completa un lightbox para una visualización más detallada.

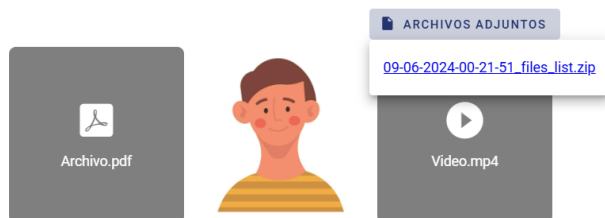


Figura 15: Componente Gallery

Fuente: Elaboración propia

3.6.2. Vistas Generales

Inicio

En la página de inicio se encuentra la bienvenida al sistema y una pequeña explicación de cómo utilizar el componente **Selector Feria de Software Global**, además de un botón que redirige a la página oficial de la Feria de Software.

Más abajo se encuentra un buscador el cual permite realizar búsquedas sobre los elementos que representan a la Feria de Software, cada una de las ferias mostrará su información en el componente **FeriaCard**. Además esta vista cuenta con una paginación la cual permite consultar en pequeñas porciones la información lo que ayuda a un rendimiento mucho más óptimo, ya que no trae toda la información de la base de datos.



Figura 16: Vista Inicio

Fuente: Elaboración propia

- **FeriaCard:** Componente en el cual muestra toda la información asociada a una Feria de Software, se compone de 3 partes principales:
 - **Portada:** En la parte superior del componente se encuentra la imagen principal que representa a la Feria de Software, además de su información básica, como lo es, su edición, fecha y lugar del evento.
 - **Accesos directos:** La segunda parte del componente tiene accesos directos a los tipos de contenidos relacionados a la Feria de Software, al hacer clic en alguno de estos elementos se llevará al usuario a la página correspondiente, solo que filtrando el contenido a esa Feria en específico, cabe recalcar que el comportamiento no es el mismo que el de **Selector Feria de Software Global**.
 - **Contenido:** Por último se encuentra la sección de contenido, la cual contiene un mensaje o descripción que se le da a la Feria de Software y todos los archivos multimedia que se adjuntaron como propios del evento, en este caso se hace uso del componente **Galería** para visualizar los archivos.



Figura 17: Componente FeriaCard

Fuente: Elaboración propia

Hitos Históricos / Historia

En esta vista se encuentran todos aquellos hitos históricos que han marcado la trayectoria de la Feria de Software, el contenido se muestra de manera descendente según el año, es decir desde los hitos más actuales a los más antiguos.

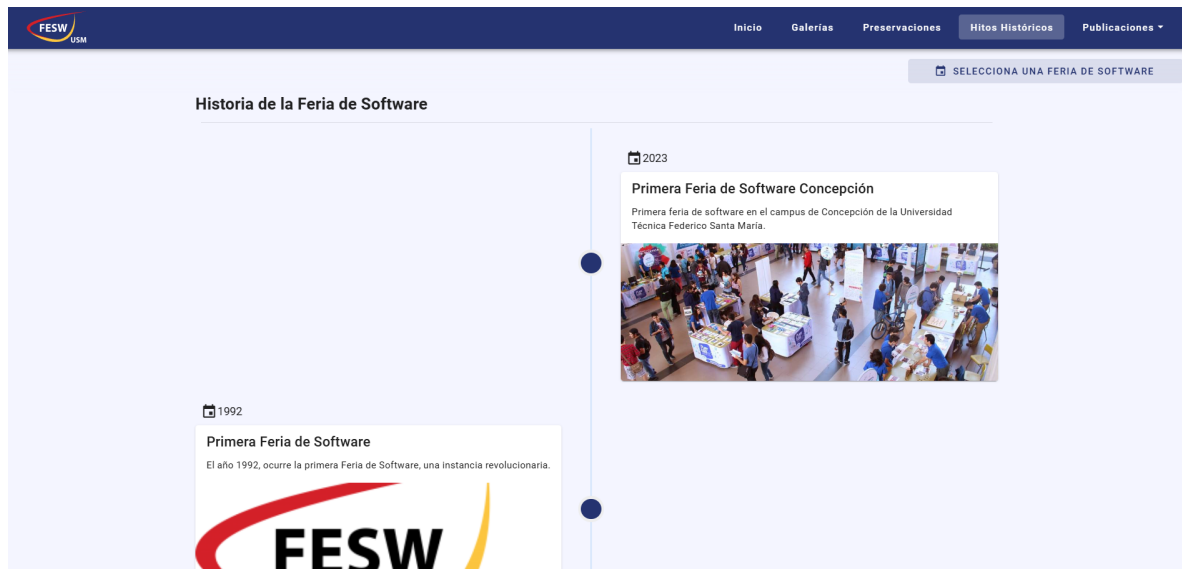


Figura 18: Vista Hitos Históricos

Fuente: Elaboración propia

- **HitosHistoricosCard:** Componente encargado de mostrar un Hito Histórico, se estructura de la siguiente forma:

- Año del hito histórico.
- Título asociado.
- Descripción del Hito Histórico.
- Imagen representativa.



Figura 19: Componente HitosHistoricosCard

Fuente: Elaboración propia

Preservaciones

La vista de preservaciones es la encargada de mostrar cada una de los contenidos de tipo **preservación**, esta vista cuenta con paginación, por lo que ayuda para un mejor rendimiento. Cada preservación se muestra utilizando el componente **PreservacionCard**.

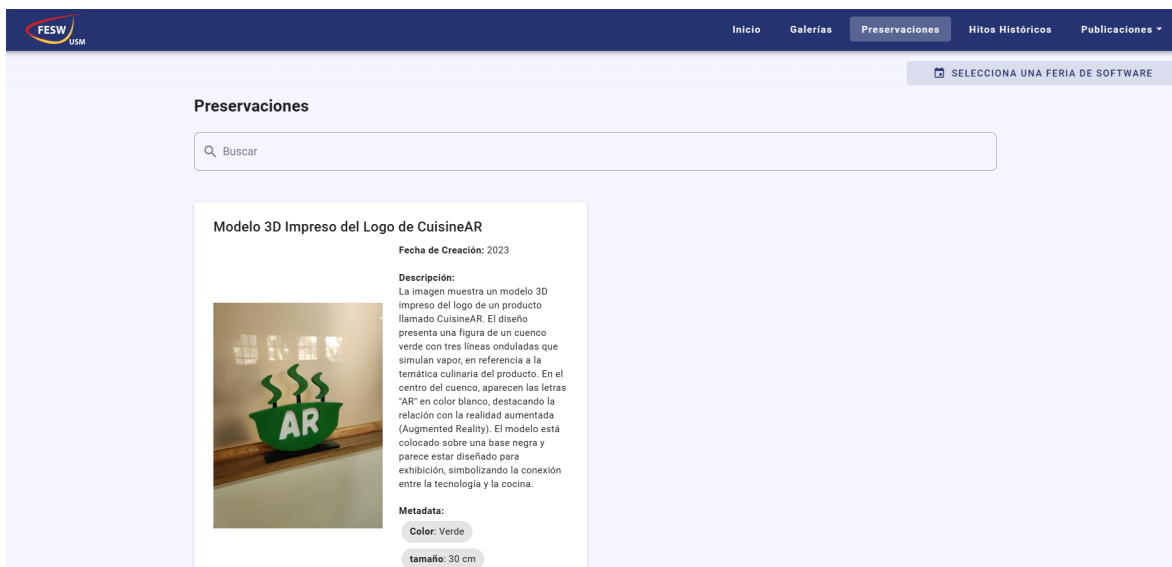


Figura 20: Vista Hitos Históricos

Fuente: Elaboración propia

- **PreservacionCard:** Componente encargado de mostrar la información relacionada a una preservación, a continuación un listado de la estructura del componente:
 - Título de la preservación.
 - Carousel de contenidos multimedia, donde se puede ver todos los archivos asociados a una preservación, como lo son imágenes, vídeos, archivos, entre otros.
 - Año de creación.
 - Descripción del elemento a preservar.
 - Metadata de preservación, donde se encuentra toda aquella información intrínseca del elemento a preservar.

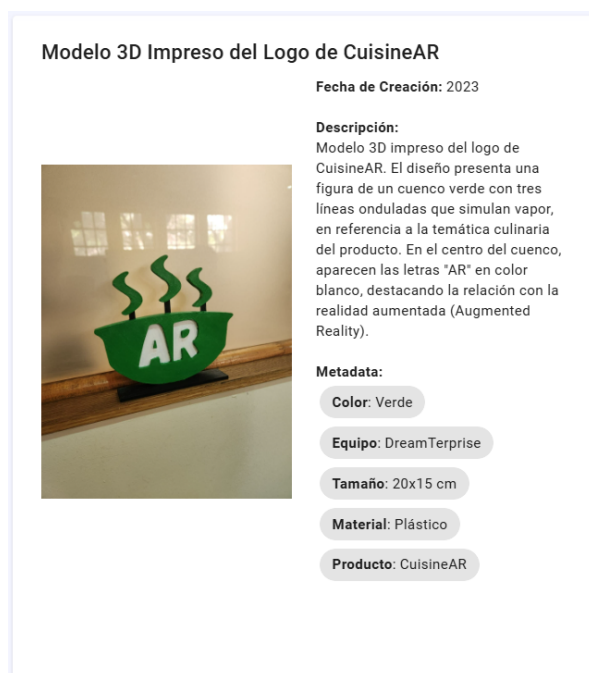


Figura 21: Componente PreservacionCard

Fuente: Elaboración propia

Galerías

En la sección de galerías se encuentran aquellos contenidos multimedia, tales como fotografías y vídeos. Cada tipo de contenido es presentado de manera accesible y visualmente atractiva, permitiendo a los usuarios explorar el material histórico de forma sencilla. Para representar cada uno de estos tipos de contenido, se utiliza una tarjeta denominada **GaleríaCard**.

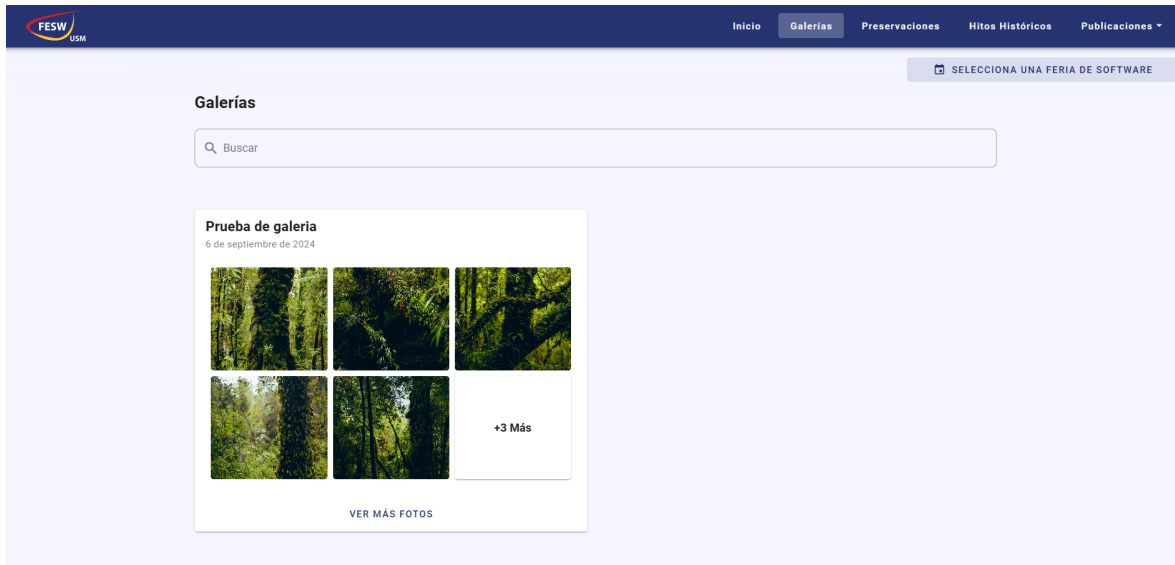


Figura 22: Vista Galerías

Fuente: Elaboración propia

- **GaleríaCard:** Componente encargado de representar una publicación de tipo galería, esta tarjeta contiene información parcial del contenido, como lo es, el título, la fecha y una vista previa de algunos contenidos multimedia que incluye, además de información que indica la cantidad de archivos multimedia adicionales que contiene la galería. Por último contiene un botón con el texto "VER MÁS FOTOS" el cual al ser pulsado redirige a la vista individual de la galería donde se podrá ver con más detalle toda la información de la publicación.



Figura 23: Componente GaleriaCard

Fuente: Elaboración propia

Investigaciones

Vista encargada de mostrar todos los contenidos de tipo investigaciones, cada uno de los elementos es representado por el componente **InvestigacionesCard**.

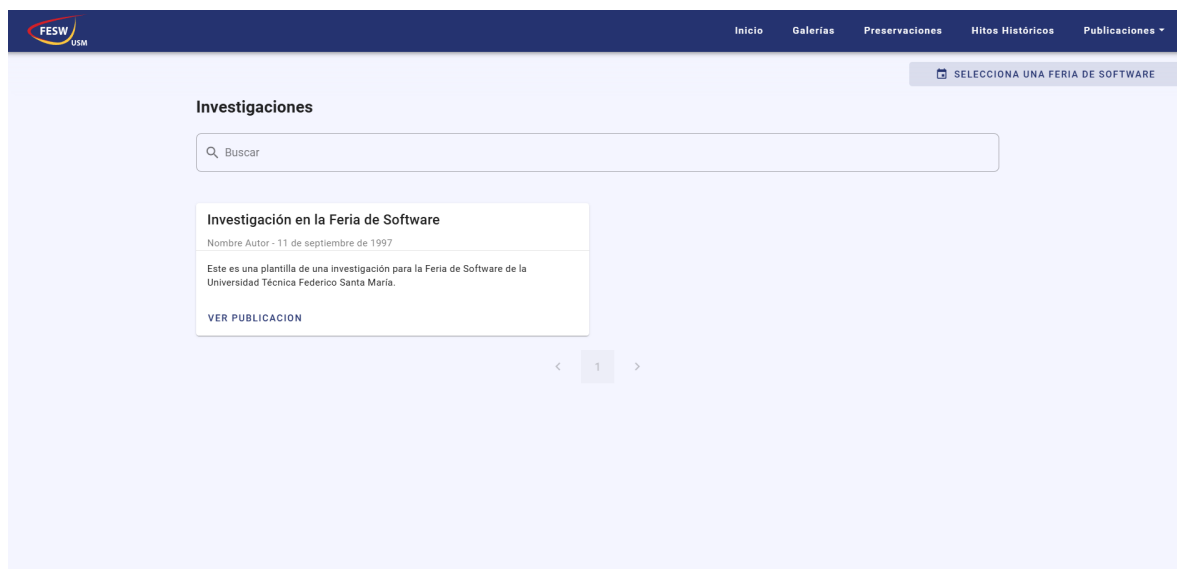


Figura 24: Vista Investigaciones

Fuente: Elaboración propia

- **InvestigacionesCard:** Componente encargado de mostrar la información parcial de las Investigaciones, se estructura de la siguiente forma:

- Título de la investigación.
- Autor y fecha de publicación.
- Descripción del contenido.
- Botón para dirigirse al detalle completo de la investigación.



Figura 25: Componente InvestigacionesCard

Fuente: Elaboración propia

Proyectos Destacados

Vista encargada de mostrar todos los proyectos destacados, cada uno de estos es representado por el componente **ProyectosDestacadosCard**.

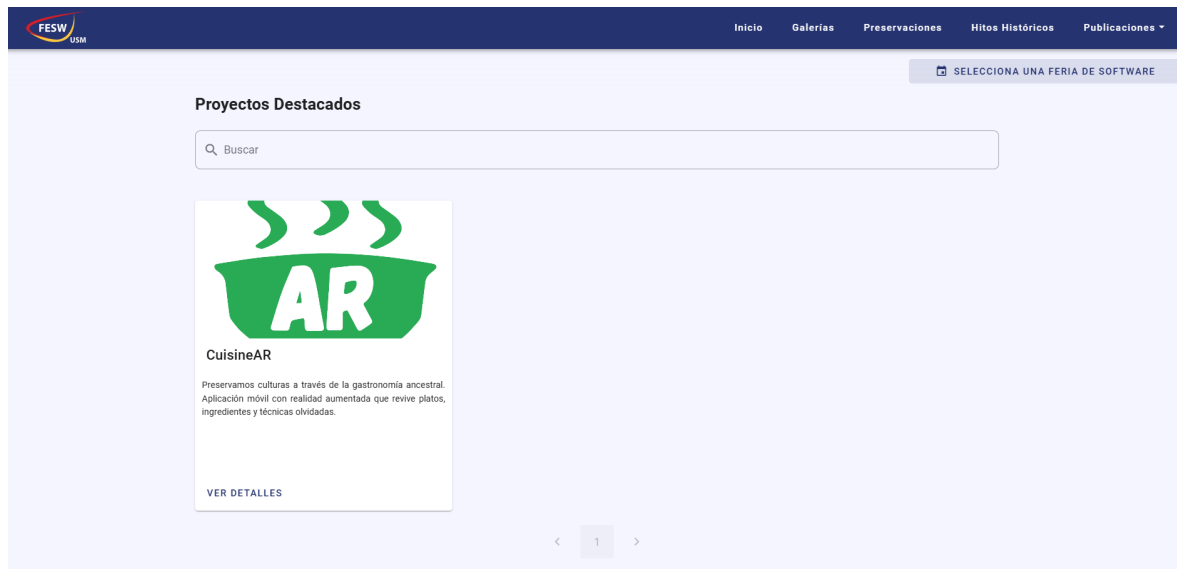


Figura 26: Vista Proyectos Destacados

Fuente: Elaboración propia

- **ProyectosDestacadosCard:** Componente encargado de representar un Proyecto Destacado, su estructura es de la siguiente manera:
 - Imagen representativa del Proyecto Destacado.
 - Nombre/título de la publicación del Proyecto Destacado.
 - Descripción del Proyecto Destacado.
 - Botón para poder acceder al detalle completo del Proyecto Destacado.



Figura 27: Componente ProyectosDestacadosCard

Fuente: Elaboración propia

Impacto en los Medios

Vista encargada de mostrar las publicaciones de tipo Impacto en los Medios; se representa por el componente **ImpactoMediosCard**.



Figura 28: Vista Impacto en los Medios

Fuente: Elaboración propia

- **ImpactoMediosCard:** Componente encargado de representar a una publicación de tipo Impacto en los Medios, su estructura es de la siguiente manera:

- Título del Impacto en los Medios.
- Autor y fecha.
- Descripción de la publicación.
- Carrousel de archivos multimedia adjuntos.
- Botón para poder acceder al detalle completo de la publicación.



Figura 29: Componente ImpactoMediosCard
Fuente: Elaboración propia

Testimonios

Esta vista se encarga de mostrar todos los testimonios que están disponibles en la aplicación, cada uno de estos es representado por el componente **TestimoniosCard**.

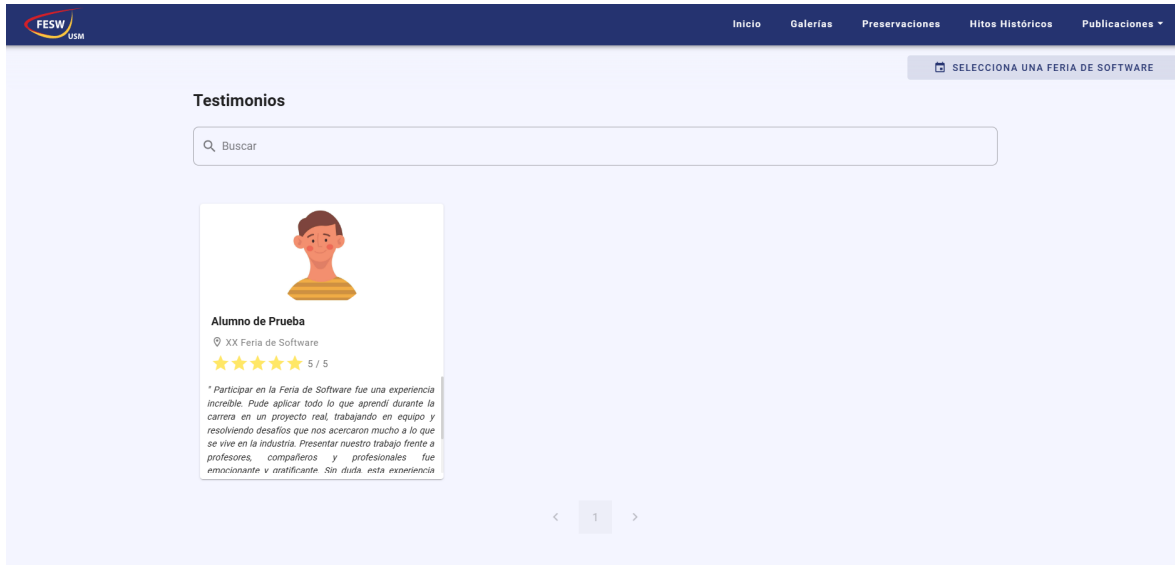


Figura 30: Vista Testimonios

Fuente: Elaboración propia

- **TestimoniosCard:** Componente encargado de mostrar los contenidos de tipo Testimonio, la estructura con la que se presenta es la siguiente:
 - Imagen de el o los autores del testimonio.
 - Nombre de el o los autores del la publicación.
 - Edición de la Feria de Software asociado al testimonio.
 - Calificación asignada a la Feria de Software por parte del autor, representado con estrellas en una escala de 1 - 5.
 - Contenido del testimonio, el cual tiene un comportamiento para poder adaptarse y navegar de manera adecuada.



Figura 31: Componente TestimoniosCard

Fuente: Elaboración propia

3.6.3. Vistas Administración

La vista de Administración es el lugar donde se gestionan todos los contenidos de la aplicación. En esta sección, los administradores tienen la capacidad de crear, editar o eliminar cualquier contenido disponible. La estructura de la vista incluye un menú conocido como **Navigation Drawer**, ideal para este tipo de interfaz, ya que ofrece acceso rápido y sencillo a todos los apartados de la aplicación.

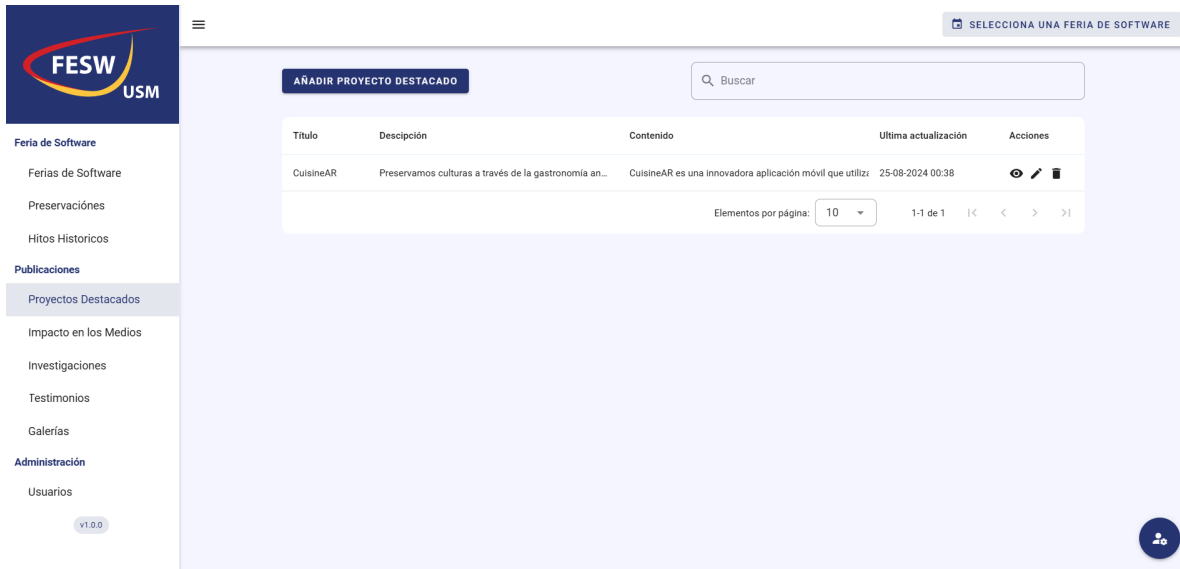


Figura 32: Vista Administrador

Fuente: Elaboración propia

Además, se dispone de una tabla donde se listan los contenidos, lo que facilita su gestión al permitir filtrar por parámetros específicos. Cada contenido se presenta en un formato tabular, lo que estandariza su administración y añade botones de acción para modificarlos o eliminarlos. La tabla también incluye paginación, optimizando la carga de los datos.

En la parte superior de la vista, se encuentra un botón que permite crear nuevos contenidos, dirigiendo al usuario a los formularios correspondientes. También se ha incorporado una barra de búsqueda para que el administrador pueda filtrar los contenidos de manera más precisa.

AÑADIR FERIA DE SOFTWARE

Q Buscar

nombre	Descripción	Fecha del evento	Ultima actualización	Acciones
XXXII Feria de Software	Descripción Feria de Software XXXII	15-11-2024	04-09-2024 23:58	 
XXXI Feria de Software	Descripción Feria de Software XXXI	10-11-2023	25-08-2024 00:16	 
XXX Feria de Software	Descripción Feria de Software XXX	25-11-2022	25-08-2024 00:01	 
XXIX Feria de Software	XXIX Feria de Software	18-03-2022	26-08-2024 12:12	 
XXVIII Feria de Software	XXVIII Feria de Software	17-12-2020	26-08-2024 12:11	 
XXVII Feria de Software	XXVII Feria de Software	13-12-2019	26-08-2024 12:11	 
XXVI Feria de Software	XXVI Feria de Software	30-11-2018	25-08-2024 00:08	 
XXV Feria de Software	XXV Feria de Software	10-11-2017	26-08-2024 12:11	 
XXIV Feria de Software	XXIV Feria de Software	18-11-2016	26-08-2024 12:11	 
XXIII Feria de Software	XXIII Feria de Software	18-03-2016	26-08-2024 12:11	 

Elementos por página: 10 1-10 de 13 |< < > >|

Figura 33: Tabla de Información vista Administración
Fuente: Elaboración propia

CAPÍTULO 4

VALIDACIÓN DE LA SOLUCIÓN

4.1. Digitalización, Recopilación y Almacenamiento de Contenidos

Parte importante del proceso de validación fue probar directamente la aplicación, para lo cual se definieron las siguientes tareas:

- Se digitalizaron y recopilaron alrededor de **230** contenidos multimedia (imágenes, folletos, afiches, entre otros).
- Se añadieron a la aplicación aproximadamente **50** publicaciones de diferentes tipos:
 - Ediciones de la Feria de Software.
 - Proyectos destacados.
 - Materiales relacionados con la preservación de la historia de la Feria de Software.
- Como parte del proceso de validación:
 - Se utilizó la aplicación para almacenar y gestionar las diferentes publicaciones, además de sus archivos multimedia asociados.
 - Se comprobó que los archivos multimedia se guarden de forma correcta en el Object Store (MinIO).
 - Se verificó que los usuarios puedan acceder a los contenidos de manera eficiente y fluida.

Este proceso de validación fue súper importante para garantizar que la solución desarrollada cumplía con sus funcionalidades claves, como lo son la de poder preservar y difundir toda la información que se le va proporcionando, además de comprobar que la aplicación funciona correctamente.

4.2. Pruebas de Usuario

Es fundamental validar la solución propuesta con usuarios, ya que esto permite analizar la interacción con la aplicación en un entorno real. Para ello, se realizaron **Pruebas de Usuario** con el objetivo de evaluar la experiencia y funcionalidad de la plataforma. Estas pruebas se llevaron a cabo en tres modalidades: algunas en el **Lab UX** de la universidad, otras de manera presencial, y algunas online debido a la disponibilidad de las partes involucradas en este proceso.

El perfil de los usuarios de prueba es importante en el contexto del proyecto, todos los usuarios están vinculados a la universidad, incluyendo alumnos de diferentes carreras y también personas fuertemente relacionadas a la Feria de Software, esta variedad permitió obtener distintas perspectivas sobre el uso de la aplicación. La edad de los participantes se encuentra entre los **22 y 35 años**.

Se elaboró un plan para realizar las pruebas necesarias, definiendo 10 pruebas claves que abarcan funcionalidades críticas de la aplicación. Estas pruebas se dividen en dos partes:

- **Pruebas para un Usuario Normal:** Se centran principalmente en la navegación a través de los diversos contenidos de la aplicación. El objetivo es analizar cómo los usuarios interactúan con la plataforma.
- **Pruebas para un Usuario Administrador:** Estas pruebas se enfocan en la gestión de contenidos, permitiendo evaluar cómo los administradores pueden crear, modificar y gestionar la información que se presenta en la aplicación.

A continuación, se detallan las 10 pruebas específicas que se realizaron:

Usuario Normal:

1. Quiero que me digas la fecha de la feria de software con el siguiente título “**XX Feria de Software**”, mostrando la información que tiene adjunta y también sus tipos de contenido.
2. Quiero que me muestres el PDF adjunto a la memoria o investigación, del autor **Joaquín Francisco Rojas Rodríguez**, la cual surge de un proyecto de la Feria de Software.

3. Dentro de la aplicación se tiene información respecto a hechos históricos de la feria de software, quiero que me digas qué ocurrió en **1992**.
4. Quiero que me muestres todos los elementos de preservación de la feria de software, una vez los encuentres, quiero que solo se muestren los elementos de preservación de la feria de software que tiene como título "**XXIX Feria de Software**", la cual ocurre en el año **2022**.
5. Quiero que me muestres todos los proyectos destacados de la feria de software que tiene como título "**30° Feria de Software**" en el año **2022**.

Usuario Administrador:

6. Ahora eres un usuario Administrador, quiero que accedas a la parte de administración desde la página de inicio en la que te encuentras actualmente.
7. Quiero que crees una feria de software con la siguiente información:

- Título: Feria Test
- Descripción: Descripción de la Feria Test
- Fecha: 18 – 12 – 2000
- Lugar: Universidad Santa María.

Además, quiero que adjuntes los archivos **imagen1.jpg** e **imagen2.jpg**, pero quiero que **imagen2.jpg** sea la portada (estos archivos se encuentran disponibles en el escritorio, en una carpeta llamada "Feria").

8. Quiero que crees una Preservación con la siguiente información:

- Título: Preservación Prueba
- Descripción: Preservación Prueba
- Fecha Creación: 2000
- Edición Feria de Software: Feria Test
- Metadata:
 - Equipo: Equipo de prueba

- Color: Color de Prueba

Agregar los archivos **preservacion1.jpg** y **preservacion2.pdf** (estos archivos se encuentran disponibles en el escritorio, en una carpeta llamada “Preservación”). Una vez creada quiero que la puedas visualizar desde la vista de administración.

9. Quiero que modifiques la Preservación que creaste anteriormente con la siguiente información:

- Título: Preservación Prueba - Editada
- Metadata:
 - Color: Color editado

Elimine el archivo **preservacion1.jpg**

10. La última tareas que quiero que realices es eliminar los siguientes contenidos, en este orden:

- Testimonio con el título “**Testimonio para Borrar**”.
- La feria de software que creaste anteriormente con título “**Feria Test**”.

Una vez eliminados los contenidos anteriores, quiero que puedas cerrar la sesión del administrador.

Para cuantificar la usabilidad de la aplicación, se aplicó el método **System Usability Scale (SUS)**, una herramienta ampliamente utilizada para medir la usabilidad de cualquier sistema. Este método proporciona una evaluación sencilla y efectiva de la experiencia del usuario, permitiendo identificar puntos de mejora y confirmar que la solución cumple con los estándares de usabilidad esperados. [Brooke, 1995]

Las preguntas son las siguientes:

- **P1:** Creo que me gustaría utilizar este sistema con frecuencia
- **P2:** Encontré el sistema innecesariamente complejo
- **P3:** Pensé que el sistema era fácil de usar

- **P4:** Creo que necesitaría el apoyo de un técnico para poder utilizar este sistema
- **P5:** Encontré que las diversas funciones de este sistema estaban bien integradas
- **P6:** Pensé que había demasiada inconsistencia en este sistema
- **P7:** Me imagino que la mayoría de la gente aprendería a utilizar este sistema muy rápidamente
- **P8:** Encontré el sistema muy complicado de usar
- **P9:** Me sentí muy seguro usando el sistema
- **P10:** Necesitaba aprender muchas cosas antes de empezar con este sistema

Estas preguntas deben ser evaluadas por el usuario utilizando la siguiente escala:

- 1: Totalmente en desacuerdo
- 2: En desacuerdo
- 3: Neutral
- 4: De acuerdo
- 5: Totalmente de acuerdo

Usuario	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	SUS
Usuario 1	4	2	4	2	3	2	4	2	3	3	67.5
Usuario 2	5	1	4	4	5	1	5	1	5	1	90
Usuario 3	4	2	5	3	5	1	5	2	5	2	85
Usuario 4	4	2	4	1	4	4	4	3	3	1	70
Usuario 5	5	2	4	1	5	1	5	1	4	1	92.5
Usuario 6	5	1	5	1	5	1	5	1	5	1	100
Usuario 7	3	1	5	1	5	1	4	2	5	1	90
Usuario 8	4	3	4	2	4	2	5	3	4	2	72.5
Usuario 9	3	2	5	1	4	3	4	2	5	2	77.5
Usuario 10	5	1	4	2	5	1	5	1	4	3	87.5
Usuario 11	4	2	5	1	3	2	4	2	5	2	80
Usuario 12	5	1	4	2	4	1	5	1	4	3	85
Usuario 13	3	2	5	1	4	3	4	3	5	2	75
Usuario 14	4	1	5	2	5	1	5	2	4	3	85

Tabla 1: Resultados System Usability Scale.

Fuente: Elaboración Propia.

Una vez aplicado el formulario SUS, se logra obtener un promedio de **82,7** para todos los usuarios. Este puntaje indica que la aplicación es usable, ya que, en comparación con estudios previos realizados en más de 500 sitios web y aplicaciones, el promedio de usabilidad fue de **68** [Jeff Sauro, 2011]. Por lo tanto, estar sobre este promedio indica que la aplicación cumple muy bien en términos de usabilidad.

4.3. Ajustes en la Aplicación

Una vez realizadas las pruebas con usuarios se puede evidenciar comportamientos que no están del todo bien definidos, por lo que es importante abordarlos y generar un producto de software de calidad, en el contexto del proyecto se identificaron 2 puntos importantes a los que se realizaron mejoras:

- Compresión de imágenes, dado que la mayoría de los contenidos y publicaciones están representados con imágenes, se evidenció que es uno de los formatos que más se almacenarán dentro de la aplicación, por lo que cada vez que se sube una nueva imagen pasará por un proceso de compresión que asegura una calidad adecuada para su visualización, pero reduciendo su tamaño hasta en un 80 %.
- Selector de fechas perfil administrador, en las pruebas de usuarios se evidencia un comportamiento no esperado al seleccionar una fecha en los formularios, por lo que se realizan mejoras en este componente para que se pueda ingresar de forma mucho más clara y no llevar a posibles errores.

CAPÍTULO 5

CONCLUSIONES

5.1. Conclusiones Generales

En esta memoria se abordó un desafío crítico que ha vivido la Feria de Software de la Universidad Técnica Federico Santa María, la falta de un mecanismo o metodología que permita gestionar y preservar toda la información generada por más de 30 años.

La Feria de Software, año a año genera gran volumen de información, tanto en formato físico como digital, sin una forma centralizada para organizarla se generan complicaciones para acceder a ella y su correcta preservación.

Al analizar esta situación se evidencia que la información al estar tan dispersa y sin un orden, ha generado pérdidas de materiales valiosos para el legado de la Feria de Software, por lo que se plantea el desarrollo de una aplicación web con la capacidad de poder gestionar contenidos, no solo centralizando la información sino que proveyendo facilidades para su difusión quedando a disposición de toda la comunidad.

Además es fundamental mencionar que el desarrollo de la solución no fue abordado solo desde necesidades tecnológicas, sino también de una sólida base conceptual que permitió estructurar de manera eficiente el proyecto. En el marco conceptual se abordaron temas como el **OAIS (Open Archival Information System)**, ayudando a comprender el desafío de la preservación en la era digital, donde es fundamental asegurar la integridad y el acceso a los datos a lo largo del tiempo.

Aplicar el modelo OAIS permitió gestionar y estructurar la información de la Feria de Software de una forma más centralizada para asegurar su preservación, dando apoyo a generar una estructura que asegura una solución efectiva en el almacenamiento, acceso y organización de grandes volúmenes de datos.

Para el desarrollo de la aplicación se han utilizado metodologías tanto para la preservación de contenidos de forma digital como para el proceso de desarrollar software, esto ayuda a construir una solución sobre una base sólida sin tener que "reinventar la rueda", sino que

aplicando buenas prácticas y estándares ya establecidos.

En el proceso de desarrollo de la aplicación, se definen dos etapas claves para poder asegurar su escalabilidad y mantenimiento a lo largo del tiempo, la primera etapa es la creación del back-end y, en segundo lugar, el desarrollo de la interfaz, la cual sirve para visualizar y gestionar la información que se envía al back-end. Se destaca de estas etapas que:

- **Desarrollo de la API:** Se crea una API Rest para la estructura del back-end, esta es clave para garantizar que la aplicación sea escalable y mantenible a lo largo del tiempo, además se implementa un Object Store para la gestión y almacenamiento de todos los contenidos multimedia, proporcionando una solución robusta y eficiente para el manejo de grandes volúmenes de información, garantizando un orden y la capacidad de gestionar el crecimiento continuo al que se enfrenta.
- **Desarrollo de la Interfaz:** Se hace uso de Vue.js para el desarrollo del front-end, mediante el uso de componentes se estandariza cómo se muestra la información, además de un panel administrativo para gestionar todos los contenidos, también se hizo un proceso de validación donde se sometió a pruebas la aplicación con usuarios reales, para asegurar su usabilidad y su correcto funcionamiento.

La combinación de metodologías y tecnológicas modernas, da como resultado una aplicación web capaz de gestionar y difundir el legado de la Feria de software de una forma sencilla y centralizada aportando un gran valor.

5.2. Cumplimiento de los objetivos

Se plantearon 4 objetivos específicos para el desarrollo de este proyecto, estos ayudan a poder determinar el cumplimiento de la memoria:

- **"Analizar la situación actual del legado de la Feria de Software y cómo se ha gestionado"**, este objetivo se cumple con la primera parte del proyecto, donde se logra analizar el problema y evidenciar que la falta de una estrategia clara y eficiente para poder gestionar todos los contenidos generados por la Feria de Software ha llevado a una acumulación de información sin ningún orden, provocando pérdidas.

- **"Proponer una solución para manejar el legado de la Feria de Software"**, este segundo objetivo se cumple con la investigación de mecanismos y tecnologías adecuadas para dar solución al problema y plantear una solución utilizando estas tecnologías que se adaptan de la mejor forma al contexto específico.
- **"Implementar la solución propuesta para mantener y difundir el legado de la Feria de Software"**, este objetivo se cumple con la implementación de la solución, utilizando las tecnologías, no solo de forma conceptual sino que a nivel técnico, desarrollando la aplicación y dejándola operativa en un entorno de producción en los servidores del Departamento de Informática.
- **"Verificar el uso de la solución y realizar ajustes necesarios"**, este objetivo se cumple en la última parte del desarrollo de la memoria, una vez llevada la aplicación a un entorno de producción se utiliza para subir contenidos, además de hacer pruebas con usuarios reales para asegurar su correcto funcionamiento y usabilidad.

Con el cumplimiento de los objetivos anteriormente mencionados se puede dar por completado todo el proceso planteado, por lo que finalmente se logra cumplir el objetivo general: **"Análisis y desarrollo de una aplicación web para la Difusión y Preservación del legado generado por la Feria de Software del Departamento de Informática en la Universidad Técnica Federico Santa María"**.

5.3. Trabajo Futuro

A medida que pasa el tiempo es posible enfrentar nuevos desafíos o mejoras que se pueden realizar en la aplicación, por lo que algunas ideas son:

- Agregar nuevos tipos de contenidos en la aplicación a medida que la Feria de Software lo vaya necesitando, pueden surgir nuevas iniciativas que no se contemplaron en esta memoria y, de esta forma, seguir preservando el legado de esta actividad.
- Integración con Inteligencia Artificial (IA), de modo de facilitar la creación de los contenidos, se podría realizar una integración donde una IA pueda tomar las imágenes,

archivos, entre otros contenidos multimedia y, de esta forma, completar automáticamente la información solicitada por cada formulario de la aplicación, facilitando la creación de contenidos. En la Figura 34 se puede ver una prueba que se realizó, pasando como entrada una imagen de una tarjeta de un equipo que participó en la Feria de Software, de esta forma se evidencia lo factible de la integración.

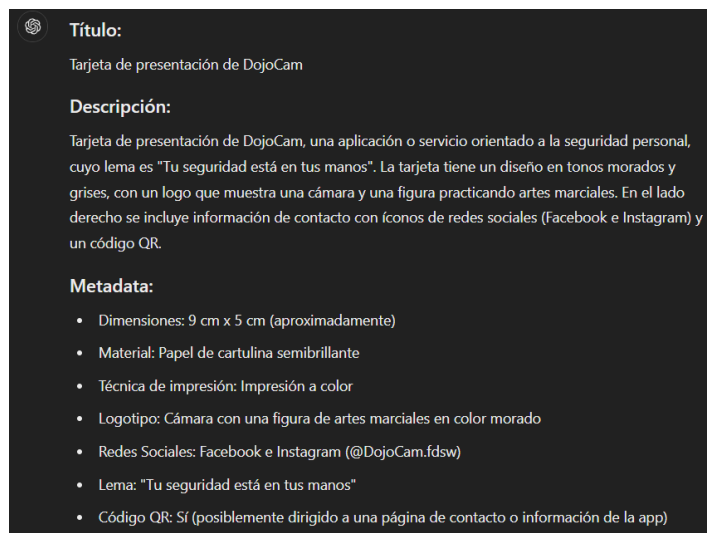


Figura 34: Respuesta a ChatGPT

Fuente: Elaboración propia

- Buscador global de contenidos, esto permite que los usuarios que utilicen la aplicación puedan encontrar de forma mucho más fácil y directa cualquier tipo de contenido solo realizando una búsqueda.
- Filtros avanzados para contenidos, poder implementar filtros avanzados permite ser más preciso con la búsqueda y los diferentes tipos de contenidos.

5.4. Palabras finales del autor

Realizar esta memoria fue un gran desafío, comenzar un proyecto desde el punto cero, es decir desde la definición del problema, investigar cómo se han abordado problemáticas similares, pero en otros contextos, hasta el desarrollo y puesta en producción de la solución, fue un proceso largo y lleno de aprendizajes.

A lo largo del proyecto me encontré con muchas dificultades y momentos de frustración, en los que me sentía abrumado, pero al mismo tiempo esto ayudó a seguir adelante, ver cómo todo se iba armando poco a poco y cada una de sus partes se conectaban entre sí, me ayudó a darme cuenta de que esto es algo que realmente me gusta, pasar por todas las etapas del desarrollo de software me permitió ver de cerca lo que lleva realizar un proyecto, desde una simple idea hasta quedar operativo para el uso de la comunidad.

Enfrentar y resolver problemas, aprender de cada paso que iba dando junto al proyecto, fue importante para mi crecimiento tanto personal como profesional, estando en este punto final del proyecto y mirar hacia atrás, puedo decir que esta experiencia me ayudó a generar mayor confianza en mis conocimientos y capacidades, además de animarme a enfrentar nuevos desafíos en mi futuro entorno profesional.

REFERENCIAS BIBLIOGRÁFICAS

- [Amodeo, 2013] Amodeo, E. (2013). Principios de diseño de apis rest. *Canada: Leanpub*.
- [Boté-Vericad, 2012] Boté-Vericad, J.-J. (2012). Propuesta de un modelo de preservación digital para pequeñas y medianas instituciones sanitarias.
- [Brooke, 1995] Brooke, J. (1995). Sus: A quick and dirty usability scale. *Usability Eval. Ind.*, 189.
- [CCSDS, 2019] CCSDS (2019). Reference Model For An Open Archival Information System (OAIS). *Magenta book*.
- [Express, 2024] Express (2024). Express - Infraestructura de aplicaciones web Node.js. accedido 29-04-2024 <https://expressjs.com/es/>.
- [Factor et al., 2005] Factor, M., Meth, K., Naor, D., Rodeh, O., y Satran, J. (2005). Object storage: the future building block for storage systems. En *2005 IEEE International Symposium on Mass Storage Systems and Technology*, pp. 119–123.
- [FC Barcelona, 2024] FC Barcelona (2024). Historia del Barça | canal oficial fc barcelona. Accedido el 29-09-2024 <https://www.fcbarcelona.es/es/club/historia/decada-a-decada>.
- [Gracy y Kahn, 2012] Gracy, K. F. y Kahn, M. B. (2012). Preservation in the digital age. *Library Resources and Technical Services*, 56(1):25–43.
- [Jeff Sauro, 2011] Jeff Sauro, P. (2011). Measuring Usability with the System Usability Scale (SUS). Accedido el 04-10-2024 <https://measuringu.com/sus/>.
- [Lavoie, 2014] Lavoie, B. (2014). *The Open Archival Information System (OAIS) Reference Model: Introductory Guide (2nd Edition)*.
- [LightGallery, 2024] LightGallery (2024). Lightgallery. Accedido el 12-09-2024 <https://www.lightgalleryjs.com/>.
- [Muñoz, 2006] Muñoz, B. (2006). La gestión de riesgos orientada a la conservación de información en soporte digital. *Documentación de las Ciencias de la Información*, 29:125.

[Real Madrid CF, 2024] Real Madrid CF (2024). Real madrid cf | web oficial. Accedido el 29-09-2024 <https://www.realmadrid.com/es-ES/el-club/historia/futbol>.

[Reyes Covarrubias y Hevia Rodríguez, 2018] Reyes Covarrubias, C. y Hevia Rodríguez, L. (2018). Feria de software: una experiencia consolidada de innovación educativa. *Memorias CIIE: Tendencias Educativas, Ponencias de Innovación*, (1129).

[Departamento de Informática, 2023] Departamento de Informática (2023). Presentación del departamento. accedido 20-11-2023 <https://www.inf.utfsm.cl>.

[Ticona Condori, 2022] Ticona Condori, H. R. (2022). Desarrollo de aplicativos webs, beneficios y diferencias al utilizar un framework o un cms (content management system).

[Vue.js, 2024] Vue.js (2024). Introducción — Vue.js. accedido 29-04-2024 <https://es.vuejs.org/v2/guide/>.