



**UCSC**

**UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN  
VICERRECTORÍA ACADÉMICA  
DIRECCIÓN DE DOCENCIA  
FACULTAD DE INGENIERÍA**

**DESARROLLO DE SISTEMA DE ADMINISTRACIÓN DE CONTENIDOS PARA  
EL SITIO WEB DE BIBLIOTECA**

**DIRECCIÓN DE BIBLIOTECAS UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA  
CONCEPCIÓN**

**JOSÉ MANUEL LICANCURA DÍAZ**

Informe de Práctica Tutelada para optar al título de  
**INGENIERO CIVIL INFORMÁTICO**

**Supervisor:** Nicolás Israel Llanos Muñoz  
**Profesora Tutora:** Mariella Gutiérrez Valenzuela

Concepción, 10 de septiembre de 2024



## Resumen ejecutivo.

El presente informe contiene las descripciones de las actividades realizadas en el periodo de realización de práctica profesional tutelada en la biblioteca central de la universidad católica de la santísima concepción.

El principal objetivo fue el desarrollo de un sistema de administración de contenidos para la nueva versión del sitio web de la biblioteca, utilizando tecnologías de desarrollo modernas y prácticas de desarrollo seguro.

En la primera etapa se identificó las funcionalidades requeridas, así como también el diseño de interfaz y estructura además la seguridad del sistema, para iniciar el desarrollo se usó una plantilla de interfaz (PrimeReact | React UI Component Library, s. f.) entregada por el supervisor, se depuró y optimizó para comenzar el desarrollo, se desarrollaron formularios de carga de contenidos en ventanas modal con validaciones de campos obligatorios y validaciones de archivos, tablas de contenidos que mostraban información de registros según el módulo en que se encontraba con opciones de edición, eliminación y visualización, estos contenidos se mostraban, editaban y cargaban mediante APIS y puntos de acceso.

En etapas posteriores se realizaron mejoras visuales y de usabilidad en el sistema, como por ejemplo el recargar de la tabla sin necesidad de actualizar la página, tamaño de ventanas modal y alertas en el sistema. Todo el tiempo estas actualizaciones eran validadas por el supervisor a cargo.

Los resultados obtenidos incluyen una interfaz de usuario intuitiva, mayor eficiencia en la carga y edición de contenido, y una seguridad robusta en el acceso al sistema. Estos avances no solo han optimizado la administración del sitio de la biblioteca, sino que también han proporcionado una experiencia de usuario más amigable y segura.

La práctica profesional ha sido una experiencia enriquecedora, permitiendo la aplicación de conocimientos teóricos en un entorno real y el desarrollo de habilidades prácticas en el ámbito del desarrollo de software y la gestión de contenidos. Las reflexiones finales destacan la importancia de la planificación y la comunicación efectiva en proyectos de desarrollo, así como la necesidad de estar actualizado con las tecnologías y prácticas de seguridad más recientes.

En conclusión, la práctica profesional ha cumplido con los objetivos planteados, proporcionando resultados significativos para la organización y valiosos aprendizajes para el crecimiento profesional.



Índice de contenidos.	Página
Capítulo 1. Introducción .....	5
1.1 Organización .....	5
1.2 Objetivos generales.....	5
1.3 Objetivos específicos .....	5
1.4 Metodología del trabajo utilizada.....	6
1.5 Contribución a la organización .....	6
Capítulo 2. Antecedentes generales de la organización .....	6
2.1 Organización .....	6
2.2 Tecnologías .....	7
Capítulo 3. Descripción detallada de las actividades realizadas .....	7
3.1 Descripción de proyectos y tareas .....	7
3.2 Herramientas y tecnologías utilizadas .....	17
Capítulo 4. Resultados .....	18
4.1 Resultados obtenidos.....	18
4.2 Aprendizajes logrados .....	18
4.3 Desafíos enfrentados .....	18
4.4 Balance de lo Planificado vs. lo Realizado.....	18
Capítulo 5. Reflexión .....	19
5.1 Aprendizajes y experiencias .....	19
5.2 Habilidades CDIO.....	19
Capítulo 6. Conclusiones .....	20
6.1 Trabajo futuro .....	20
6.2 Reflexión final.....	20
Referencias .....	21



## Índice de Tablas

Tabla 3.1: Campos de entradas en formulario de boletines.

Tabla 3.2: Campos de entradas en formulario de noticias.

Tabla 3.3: Campos de entradas en formulario de libros.

Tabla 3.4: Campos de entradas en formulario de talleres.

Tabla 3.5: Campos de entradas en formulario de recursos.

## Índice de Figuras

Figura 3.1: Funcionamiento básico de un token.

Figura 3.2: Diagrama de flujo del proceso de carga de contenido.

Figura 3.3: Captura de pantalla de módulo de Talleres SIBUCSC.

Figura 3.4: Captura de pantalla del formulario de nuevo registro en el módulo de Noticias.

Figura 3.5: Captura de pantalla de formulario de tesis digitales.



## **Capítulo 1: Introducción**

El presente informe tiene como objetivo documentar y analizar las actividades realizadas durante la Práctica Profesional Tutelada en la biblioteca central de la Universidad Católica de la Santísima Concepción. Se busca ofrecer una visión detallada de las tareas desarrolladas, resultados obtenidos, y las competencias adquiridas durante este periodo.

### **1.1 Organización**

En la Unidad de Tecnologías de Información (UTI) de la biblioteca central ubicada en el campus san Andrés, se desarrolló la práctica profesional tutelada, supervisada por Nicolás Llanos Muñoz (Especialista Unidad Tecnologías de la Información), bajo la modalidad de trabajo semi presencial, por un periodo de 19 semanas. Durante estas semanas se desarrolló un sistema de gestión y administración de contenidos del nuevo sitio web de la biblioteca de la UCSC denominado SIBUCSC (<https://www.sibucsc.cl/>), además se contribuyó con tareas de mantenimiento de software de actuales sistemas.

### **1.2 Objetivos Generales**

El principal objetivo fue la creación de un sistema de administración de contenidos del nuevo portal de la biblioteca.

Como objetivos secundarios fue la contribución y mejoras en actuales sistemas utilizados por esta unidad, la cual fue específicamente en el diseño de interfaz de usuario, como se presentan los datos, y como se entrega información sensible a los usuarios.

### **1.3 Objetivos específicos**

Los objetivos específicos de la práctica profesional fueron los siguientes:

- Aplicar conocimientos teóricos adquiridos durante la carrera en un entorno real de trabajo.
- Desarrollar y mejorar el sistema de administración de contenidos de la biblioteca institucional, utilizando tecnologías modernas y prácticas de desarrollo seguro.
- Adquirir habilidades prácticas en el desarrollo de software, gestión de proyectos y trabajo en equipo.



## **1.4 Metodología del trabajo utilizada**

Se utilizó una metodología basada en la colaboración continua y el desarrollo iterativo. Las actividades principales incluyeron reuniones periódicas con el equipo de UTI y otros departamentos clave, análisis del código existente, implementación de nuevas funcionalidades, y pruebas para asegurar la calidad del producto. Además, se hizo uso de herramientas de comunicación y colaboración en línea para coordinar eficazmente las tareas, tanto de forma remota como presencial

El desarrollo del proyecto se llevó a cabo utilizando una metodología híbrida, combinando días de trabajo presencial y remoto.

- Trabajo presencial: Durante los días presenciales, se realizaron reuniones de coordinación, revisión de avances y desarrollo colaborativo en el sitio de trabajo. Estas sesiones fueron cruciales para alinear las expectativas del equipo y resolver problemas técnicos en tiempo real.
- Trabajo remoto: En los días de trabajo remoto, se continuó con el desarrollo del proyecto desde ubicaciones externas, manteniendo una comunicación constante a través de herramientas digitales. Este enfoque permitió flexibilidad en la gestión del tiempo y en la ejecución de tareas, garantizando la continuidad del proyecto sin interrupciones.

## **1.5 Contribución a la organización**

La principal contribución se centró en el desarrollo del sistema de gestión de contenidos para el nuevo sitio web de biblioteca que está en desarrollo, además se implementaron mejoras en la interfaz de usuario, se agregaron nuevas funcionalidades y se optimizó el rendimiento del sistema del nuevo sitio web.

## **Capítulo 2: Antecedentes generales de la organización**

### **2.1 Organización**

La Dirección de Bibliotecas de la UCSC, unidad dependiente de la Vicerrectoría Académica, tiene como objetivo dar respuesta a los requerimientos de información de la comunidad universitaria, como también, facilitar el desarrollo de habilidades informacionales en los estudiantes para el acceso y uso ético de la información en los usuarios que así lo requieran. Disponer de los espacios físicos y virtuales, ofreciendo de esta forma las mejores condiciones para el aprendizaje.

La misión y visión de la biblioteca junto con el organigrama y descripción de sus unidades se encuentran en la sección de “Quiénes somos” de su página web ([UCSC-SIBUCSC, s.f.](#))

## 2.2 Tecnologías

Dentro de la UTI se tiene por objetivo desarrollar implementar administrar y proponer soluciones tecnológicas, que permitan a los usuarios y administrativos optimizar su uso y acceso a los recursos. En esta unidad se desarrolló la práctica profesional tutelada, abordando principalmente el desafío de la creación de un sistema de administración de contenidos del nuevo sitio web de la biblioteca, utilizando actuales tecnologías de desarrollo de frontend REACT (React, s. f.), librería dependiente de JavaScript (JavaScript | MDN, 2023)

## Capítulo 3. Descripción detallada de las actividades realizadas

### 3.1 Descripción de proyectos y tareas

Durante el tiempo en la unidad de tecnología de la información, se trabajó principalmente como diseñador de interfaz de usuario (UI) en el proyecto “sistema de administración de contenidos para el sitio web de biblioteca”, el objetivo de este proyecto es que los contenidos del sitio web sean gestionados desde un sistema con una interfaz limpia e intuitiva para el usuario común, permitiendo a este opciones de carga, edición, eliminación y activación/ desactivación de contenidos que son visibles directamente desde el sitio de biblioteca. Todas estas acciones están sujetas a la verificación de acceso al usuario creado previamente por el jefe de UTI mediante token. (ver figura 3.1)

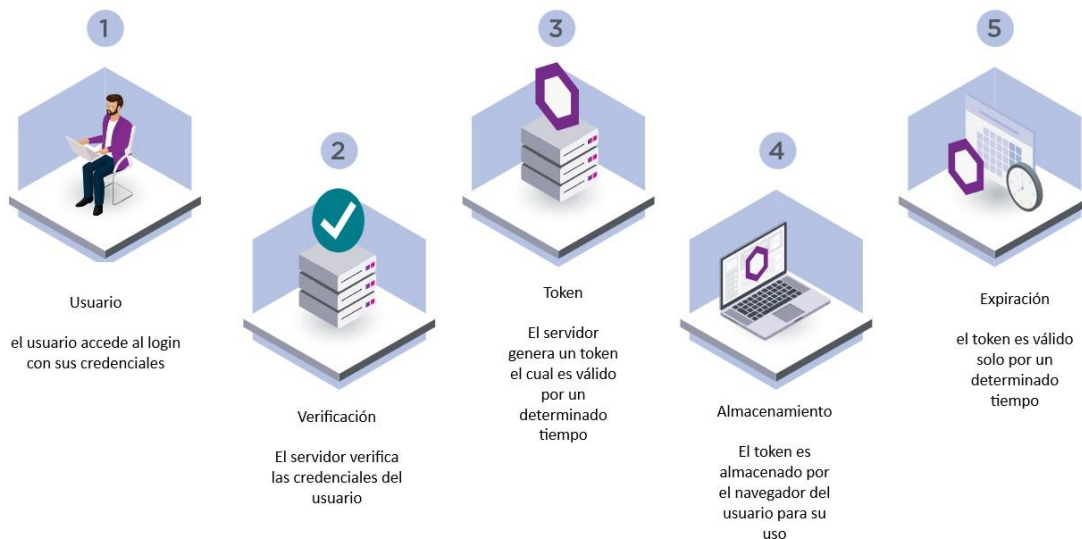


Figura 3.1 Funcionamiento básico del uso de token. Fuente: [Entrust](#).

Los módulos del sistema se describen a continuación:

a) Módulo de Boletines.

Este módulo cuenta con distintas funcionalidades, la primera es la “creación de nuevo boletín” en la cual una vez accediendo a ella mediante un botón, esta despliega en pantalla un formulario con los siguientes campos. (ver tabla 3.1)

Tabla 3.1 campos de entradas en formulario de boletines.

Campo	Tipo de Entrada	Descripción
Nombre	String	Campo de entrada para el nombre del boletín.
Número	Number	Campo de entrada para el número del boletín.
Descripción	String	Campo de entrada para la descripción del boletín.
Fecha	String	Campo de entrada para la fecha del boletín.
Imagen	File	Campo de entrada para subir una imagen (.jpg, .png, .bmp, .JPGE).
PDF	File	Campo de entrada para subir un archivo PDF (.pdf).
Campus	Select	Campo de selección con opciones cargadas desde la base de datos, representando los campus.

La imagen y el PDF envían los archivos al servidor mediante un punto de acceso creado previamente.

Los registros creados con esta funcionalidad y los existentes previamente son visualizados mediante una tabla de registros, dicha tabla tiene como encabezado los siguientes campos: nombre, numero, descripción, fecha, nombre de imagen, nombre de PDF y acciones. La tabla es creada mediante una dependencia llamada DataTables (DataTables | Javascript Table Library, s. f.) en la cual se activó la opción de búsqueda, paginación y ordenamiento.

En la columna de “acciones” cada registro tiene dos botones, editar y eliminar.

- La acción del botón de editar, abre en pantalla un formulario de edición, en el cual ya están cargados todos los campos del registro dando la opción de editar cualquier campo ya antes ingresado y guardarlo para actualizar la



información. Esta edición debe ser confirmada en el mensaje de alerta que se despliega al aceptar el formulario.

- La acción de borrar, elimina el registro seleccionado, una vez confirmada la acción mediante una alerta que se despliega en pantalla

### b) Módulo de Noticias.

Este módulo ofrece diversas funcionalidades, la principal es la "creación de una nueva noticia". Al acceder a esta opción mediante un botón, se despliega en pantalla un formulario con los siguientes campos. (ver tabla 3.2)

Tabla 3.2 campos de entradas en formulario de noticias.

Campo	Tipo de Entrada	Descripción
Tipo	Select	Campo de selección con opciones "normal" y "destacada". Cada opción determina una posición distinta en el sitio web.
Título	String	Campo de entrada para el título de la noticia.
Descripción	String	Campo de entrada para la descripción de la noticia.
Documento PDF en Enlace	File	Campo de entrada para subir un archivo PDF (.pdf).
Imagen	File	Campo de entrada para subir una imagen (.jpg, .png, .bmp, .jpeg).
Fecha de Publicación	Date	Campo de entrada para la fecha de publicación de la noticia.
Fecha de Término	Date	Campo de entrada para la fecha de término de la noticia.
Enlaces	Array	Campo de entrada para agregar múltiples enlaces relacionados con la noticia, incluyendo link y descripción.

Dentro de este formulario hay un botón que permite agregar enlaces El campo "Enlaces" es un campo dinámico el cual permite agregar desde 1 hasta 10 enlaces los cuales tienen un link y una descripción, también hay un botón para eliminar cualquiera de estos enlaces.

La imagen y el PDF son enviados al servidor mediante un punto de acceso creado previamente.

Los registros creados con esta funcionalidad, así como los existentes previamente, son visualizados en una tabla de registros, la cual tiene como encabezado los siguientes campos: tipo, título, descripción, documento PDF, imagen, fecha de publicación, fecha de término, enlaces y acciones. La tabla se crea mediante la dependencia DataTables (DataTables | Javascript Table Library, s. f.), que permite la búsqueda, paginación y ordenamiento.

En la columna de "acciones", cada registro tiene dos botones: editar y eliminar.

- **Editar:** La acción del botón de editar abre un formulario de edición en pantalla, en el cual ya están cargados todos los campos del registro, permitiendo modificar cualquier campo ingresado previamente. Una vez editado, se guarda la información y se confirma la edición mediante un mensaje de alerta que se despliega al aceptar el formulario.
- **Eliminar:** La acción de borrar elimina el registro seleccionado, después de confirmar la acción mediante una alerta que se muestra en pantalla.

c) **Módulo de Libros.**

Este módulo ofrece diversas funcionalidades, la principal es la "creación de un nuevo libro". Al acceder a esta opción mediante un botón, se despliega en pantalla un formulario con los siguientes campos. (ver tabla 3.3)

Tabla 3.3 campos de entradas en formulario de libros.

<b>Campo</b>	<b>Tipo de Entrada</b>	<b>Descripción</b>
Título	String	Campo de entrada para el título del libro.
Autor	String	Campo de entrada para el autor del libro.
Editorial	String	Campo de entrada para la editorial del libro.
Clasificación	String	Campo de entrada para la clasificación del libro.
Año	Number	Campo de entrada para el año de publicación del libro.
Imagen	File	Campo de entrada para subir una imagen (.jpg, .png, .bmp, .jpeg).



URL	String	Campo de entrada para ingresar un enlace relacionado con el libro.
-----	--------	--

La imagen y la URL son enviadas al servidor mediante un punto de acceso creado previamente.

Los registros creados con esta funcionalidad, así como los existentes previamente, son visualizados en una tabla de registros. Dicha tabla tiene como encabezado los siguientes campos: título, autor, editorial, clasificación, año, imagen, URL, y acciones.

La tabla se crea mediante la dependencia DataTables (DataTables | Javascript Table Library, s. f.), que permite la búsqueda, paginación y ordenamiento.

En la columna de "acciones", cada registro tiene dos botones: editar y eliminar.

- **Editar:** La acción del botón de editar abre un formulario de edición en pantalla, en el cual ya están cargados todos los campos del registro, permitiendo modificar cualquier campo ingresado previamente. Una vez editado, se guarda la información y se confirma la edición mediante un mensaje de alerta que se despliega al aceptar el formulario.
- **Eliminar:** La acción de borrar elimina el registro seleccionado, después de confirmar la acción mediante una alerta que se muestra en pantalla.

#### d) Módulo de talleres

Este módulo ofrece diversas funcionalidades, siendo la principal la "creación de un nuevo taller". Al acceder a esta opción mediante un botón, se despliega en pantalla un formulario con los siguientes campos. (ver tabla 3.4)

Tabla 3.4 campos de entradas en formulario de talleres.

Campo	Tipo de Entrada	Descripción
Nombre	String	Campo de entrada para el nombre del taller.
Objetivo	String	Campo de entrada para describir el propósito del taller.
Categoría	Select	Campo de selección que despliega una lista de categorías creadas previamente para el taller.
Contenido	String	Campo de entrada para detallar el material o los temas que serán abordados en el taller.



Los registros creados con esta funcionalidad, así como los existentes previamente, son visualizados en una tabla de registros. Dicha tabla tiene como encabezado los siguientes campos: nombre, objetivo, categoría, contenido, activo y acciones.

La tabla se crea mediante la dependencia DataTables (DataTables | Javascript Table Library, s. f.), que permite la búsqueda, paginación y ordenamiento.

Las categorías de los talleres son creadas, visualizadas y administradas en otra sección del módulo, estas categorías sirven como filtrado cuando el usuario realiza una búsqueda dentro del sitio web.

En la columna de "activo" hay para cada registro un switch el cual una vez pulsado este cambia de estado de activo a inactivo o de inactivo a activo según el estado inicial, el objetivo de este switch es la visibilidad del taller en el sitio web.

En la columna de "acciones", cada registro tiene dos botones: editar y eliminar.

- Editar: La acción del botón de editar abre un formulario de edición en pantalla, en el cual ya están cargados todos los campos del registro, permitiendo modificar cualquier campo ingresado previamente. Una vez editado, se guarda la información y se confirma la edición mediante un mensaje de alerta que se despliega al aceptar el formulario.
- Eliminar: La acción de borrar elimina el registro seleccionado, después de confirmar la acción mediante una alerta que se muestra en pantalla.

#### e) Módulo de recursos.

Este módulo ofrece diversas funcionalidades, siendo la principal la "creación de un nuevo recurso". Al acceder a esta opción mediante un botón, se despliega en pantalla un formulario con los siguientes campos. (ver tabla 3.5)

Tabla 3.5 campos de entradas en formulario de recursos.

Campo	Tipo de Entrada	Descripción
Nombre	String	Campo de entrada para el nombre del recurso.
Descripción	String	Campo de entrada para describir el recurso.
URL	String	Campo de entrada para ingresar un enlace relacionado con el recurso.



Imagen	File	Campo de entrada para subir una imagen (.jpg, .png, .bmp, .jpeg).
Categorías	Multiselect	Campo de selección múltiple, que se envían como un array.
Documento Externo	Checkbox/String	Al marcar esta opción, se habilita un campo adicional de tipo string para introducir una URL de un documento externo.
Documento Interno	Checkbox/File	Al marcar esta opción, se habilita un campo de tipo file para subir un archivo PDF que será almacenado en el servidor.

Las “categorías” son creadas y administradas en una sección del mismo módulo de recursos el cual permite la gestión de categorías que se visualizan al momento de crear un nuevo taller y además sirven para el filtrado de contenido cuando el usuario realiza una búsqueda.

Los registros creados con esta funcionalidad, así como los existentes previamente, son visualizados en una tabla de registros. Dicha tabla tiene como encabezado los siguientes campos: nombre, descripción, URL, imagen, categorías, visibilidad y acciones. La tabla se crea mediante la dependencia DataTables (DataTables | Javascript Table Library, s. f.), que permite la búsqueda, paginación y ordenamiento.

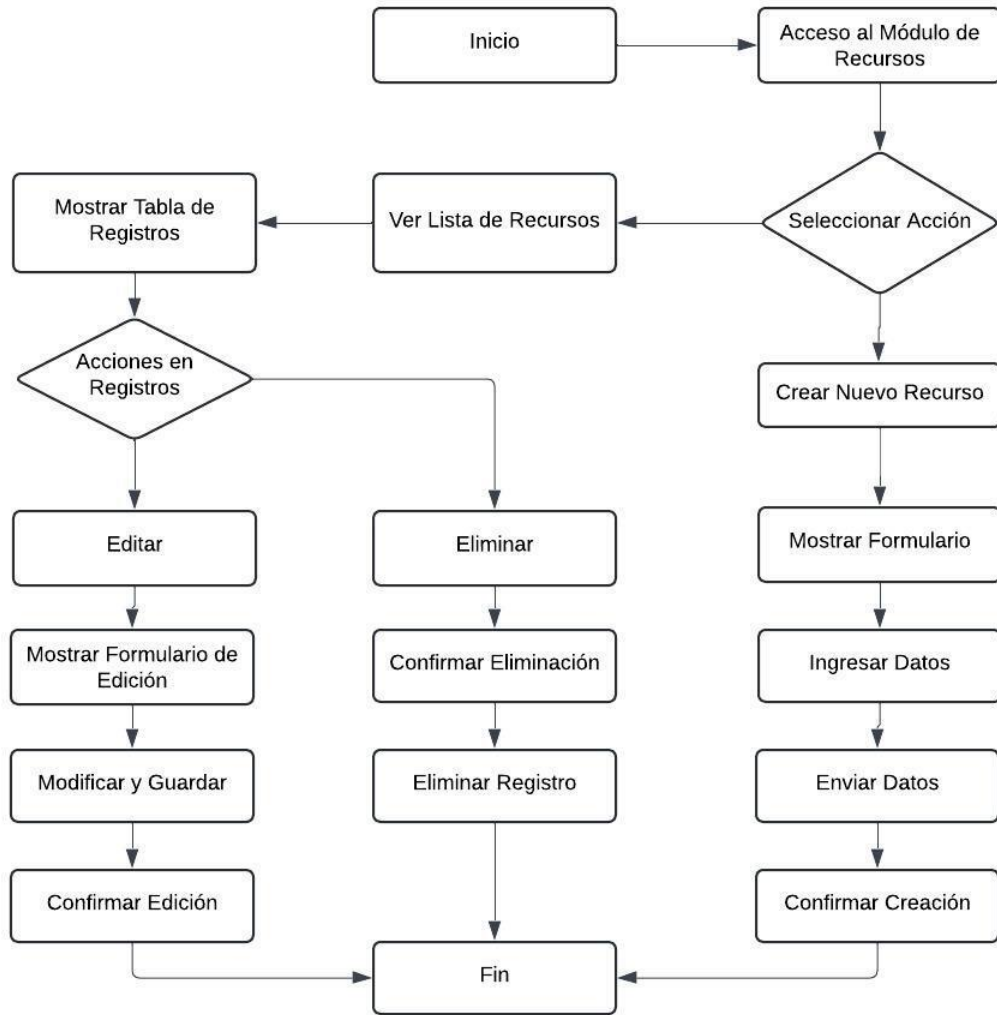
En la columna de “visibilidad” hay un switch el cual permite cambiar el estado del recurso para hacerlo visible u oculto en el sitio web

En la columna de "acciones", cada registro tiene dos botones: editar y eliminar.

- **Editar:** La acción del botón de editar abre un formulario de edición en pantalla, en el cual ya están cargados todos los campos del registro, permitiendo modificar cualquier campo ingresado previamente. Una vez editado, se guarda la información y se confirma la edición mediante un mensaje de alerta que se despliega al aceptar el formulario.
- **Eliminar:** La acción de borrar elimina el registro seleccionado, después de confirmar la acción mediante una alerta que se muestra en pantalla.



Estos módulos tienen un funcionamiento similar, se puede apreciar su comportamiento en el siguiente diagrama. (ver figura 3.2)



**Figura 3.2.** Diagrama de flujo del módulo de recursos. *Fuente: Elaboración propia.*

La interfaz de usuario de cada módulo es similar, se puede apreciar en la siguiente figura como es por ejemplo en la sección de “Talleres SIBUCSC” . (ver figura 3.3).

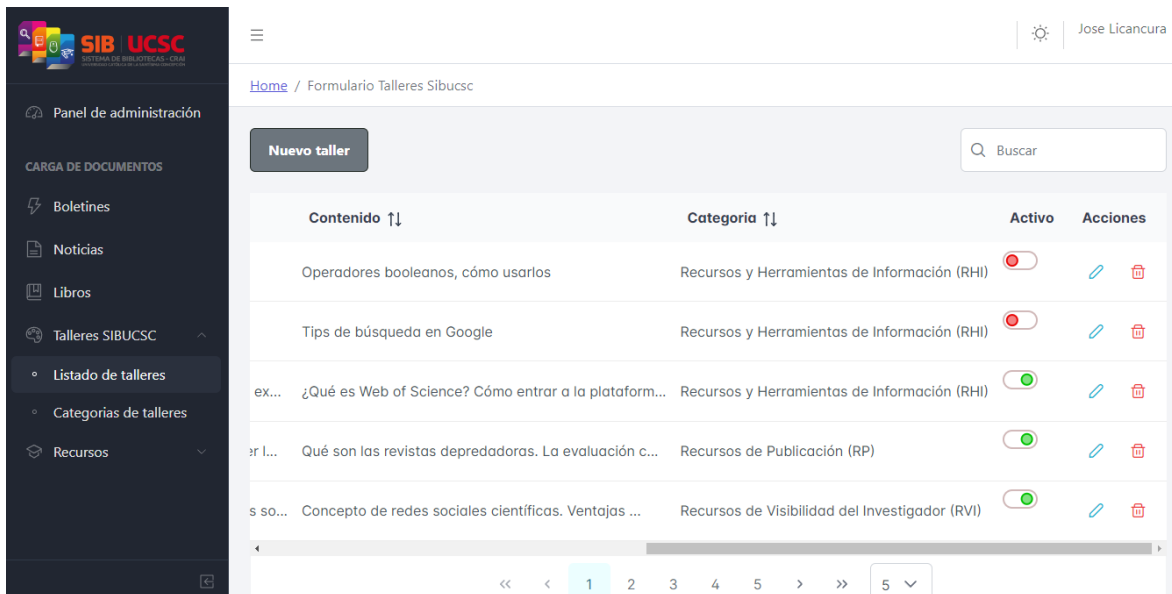


Figura 3.3 Captura de pantalla del sistema de gestión de contenidos en el módulo de talleres SIBUCSC.

La interfaz de usuario de los formularios para el ingreso de registros mantiene un diseño uniforme en todas las secciones, con ligeras variaciones en los campos de entrada según los requisitos específicos de algunos módulos. A continuación, se presenta el formulario de nuevo registro correspondiente a la sección de 'Noticias' (ver Figura 3.4).

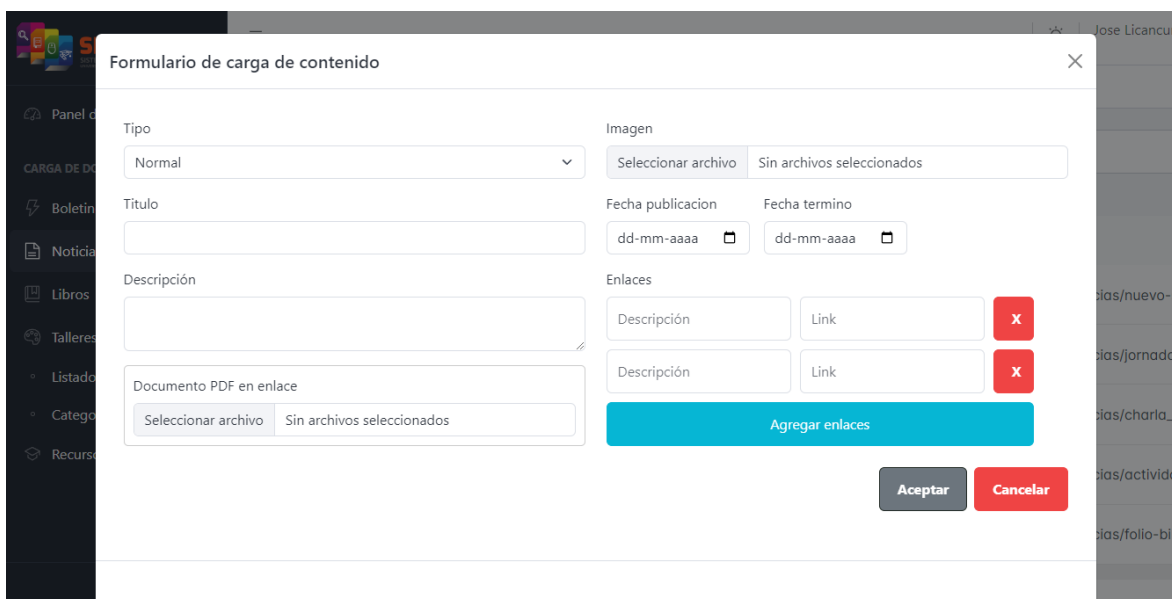


Figura 3.4 Captura de pantalla del formulario de nuevo registro en el módulo de "Noticias".



Entre otras tareas, también se abordó el mantenimiento y la actualización del sistema "MINERVA," una herramienta de gestión utilizada por el personal administrativo de la biblioteca. Dentro de este sistema, se abordó específicamente el módulo de Tesis Digitales, donde fue necesario realizar mejoras importantes. Uno de los principales cambios fue la sustitución de campos de entrada estáticos por campos dinámicos, que permiten una edición en tiempo real, mejorando así la flexibilidad y usabilidad del sistema, (ver figura 3.4).

RUT	Nombre	Email
190877345	<input type="text" value="Nicolas"/>	nllanos@ucsc.cl
161522376	Mauricio	mpenaillo@ucsc.cl

Figura 3.5 Captura de pantalla de formulario de tesis digitales

Adicionalmente, se enfrentó el desafío de desarrollar un visor para tesis digitales que no están autorizadas para su publicación. Aunque estas tesis no pueden descargarse, se requiere que estén disponibles para consulta en un formato de solo lectura. Para cumplir con este requerimiento, se implementó una solución basada en crear un dialogo IFRAME, integrado directamente en la página de búsqueda del sistema. La lógica utilizada se representa a continuación:

### Proceso de Desarrollo del Visor Seguro para Tesis Digitales

1. Solicitud de la Tesis por el Usuario:
  - El proceso comienza cuando un usuario solicita acceder a una tesis digital a través del sistema de búsqueda.
2. Servidor Encripta la Tesis PDF a Binario:
  - Una vez que se solicita la tesis, el servidor convierte el archivo PDF de la tesis en un formato binario encriptado. Esto asegura que el archivo no pueda ser fácilmente descargado o manipulado.
3. El Usuario Accede a la Página de Búsqueda:
  - El usuario navega a la página de búsqueda del sistema donde se presenta la lista de tesis disponibles para consulta.



4. Carga del Diálogo IFRAME en la Página de Búsqueda:
  - Al seleccionar una tesis, se abre un diálogo IFRAME integrado directamente en la página de búsqueda. Este IFRAME actúa como un visor en modo de solo lectura.
5. Recuperación y Descriptación del Documento en el Frontend:
  - El documento binario encriptado es recuperado desde el servidor y luego descriptado en el frontend, permitiendo su visualización en el IFRAME.
6. Visualización de la Tesis en Modo de Solo Lectura:
  - Dentro del visor, se deshabilitan de manera estricta las opciones de descarga e impresión, lo que protege la integridad del documento.
  - A pesar de que la descarga es difícil para usuarios comunes, se implementó esta capa adicional de seguridad para prevenir que usuarios técnicamente avanzados puedan acceder a los archivos sin autorización.

### 3.2 Herramientas y tecnologías utilizadas

Durante el desarrollo del proyecto, se empleó React (React, s. f.) como la tecnología principal para la construcción de la interfaz de usuario. Además, se integraron varias dependencias y bibliotecas para mejorar la funcionalidad y la experiencia del usuario:

- DataTables (DataTables | Javascript Table Library, s. f.): Utilizada para la creación y gestión de tablas con funciones avanzadas de búsqueda, paginación y ordenamiento.
- SweetAlert2 (SweetAlert2, s. f.): Implementada para mostrar alertas personalizadas y modales interactivos en la aplicación.
- React PrimiIcons (PrimeReact | React UI Component Library, s. f.-b): Biblioteca de iconos utilizada para enriquecer la interfaz con elementos gráficos.
- Axios (Axios, s. f.): Empleada para realizar solicitudes HTTP asíncronas y gestionar la comunicación con el servidor.
- PrimeReact (PrimeReact | React UI Component Library, s. f.-a): Conjunto de componentes de interfaz de usuario que facilita la creación de interfaces ricas y dinámicas.
- React DOM (ReactDOM – React, s. f.): Utilizada para manipular el DOM de manera eficiente en la aplicación React.



- Bootstrap (Bootstrap | Front-end Framework, s. f.): Framework CSS utilizado para diseñar interfaces responsivas y modernas, proporcionando una variedad de componentes y estilos predefinidos.

## Capítulo 4. Resultados

### 4.1 Resultados obtenidos

El proyecto culminó con la implementación de módulos clave como boletines, noticias, libros, talleres y recursos. Se creó la interfaz de usuario utilizando React (*React*, s. f.) y otras bibliotecas, logrando una experiencia fluida y responsiva. Además, se integraron funcionalidades para la carga y gestión de archivos y enlaces, ofreciendo a los usuarios mayor control sobre la información.

En el caso de las tesis digitales, la implementación de la funcionalidad que permite modificar, agregar o editar a los alumnos y profesores directamente desde la tabla generó un impacto significativo en la optimización del tiempo para los usuarios. Asimismo, se logró un importante avance en materia de seguridad de las tesis digitales no autorizadas gracias a la creación de un visor de PDF con restricciones, que impide la copia o descarga de documentos privados.

### 4.2 Aprendizajes logrados

Durante el desarrollo, se adquirieron habilidades de programación de JavaScript (*JavaScript* | MDN, 2023) con React (*React*, s. f.) y en la integración de componentes externos, lo que mejoró la capacidad para construir interfaces complejas. Además, la experiencia del trabajo híbrido reforzó la importancia de la planificación y comunicación efectiva.

### 4.3 Desafíos enfrentados

Uno de los principales desafíos fue la programación en JavaScript (*JavaScript* | MDN, 2023) con REACT (*React*, s. f.) ya que los conocimientos que tenía eran mínimos y desconocía REACT, pero con el autoaprendizaje y se logró que funcionara, además la adaptación a nuevos requisitos que surgieron durante el desarrollo, permitió adquirir más conocimientos lo cual era crucial para el ajuste constante de las funcionalidades. Otro reto fue mantener la consistencia en la experiencia del usuario mientras se integraban múltiples tecnologías.

### 4.4 Balance de lo Planificado vs. lo Realizado

Se logró realizar lo planificado y más, cumpliendo con todos los objetivos propuestos e implementando funcionalidades adicionales que agregaron valor al proyecto. Solo la distribución de horas sufrió algunos ajustes, principalmente para adaptarse a los desafíos y nuevos requisitos que surgieron durante el desarrollo y mantenimiento de otras plataformas.



## Capítulo 5. Reflexión

### 5.1 Aprendizajes y experiencias

El desarrollo de este proyecto ofreció valiosas oportunidades de aprendizaje y experiencias enriquecedoras en varios aspectos:

- **Dominio de Nuevas Tecnologías:** Durante el proyecto, se adquirió un dominio de React y sus bibliotecas asociadas, como DataTables, SweetAlert2, y PrimeReact.
- **Mejora en la Gestión del Tiempo:** Trabajar con plazos ajustados y un cronograma definido ayudó a mejorar las habilidades de gestión del tiempo. La capacidad para priorizar tareas, gestionar múltiples responsabilidades simultáneamente y mantener el enfoque en los objetivos principales fue un aprendizaje clave.
- **Desarrollo de Habilidades de Resolución de Problemas:** A lo largo del proyecto, surgieron desafíos técnicos y operativos que requerían soluciones rápidas y efectivas. Esto fomentó el desarrollo de habilidades para analizar problemas de manera crítica, investigar alternativas y aplicar soluciones innovadoras.
- **Adaptabilidad y Flexibilidad:** La necesidad de adaptarse a cambios en los requisitos del proyecto y a nuevas herramientas tecnológicas fortaleció la capacidad de ser flexible y receptivo.

### 5.2 Habilidades CDIO

**Concebir:** Se desarrolló la capacidad de entender y definir claramente los requisitos del proyecto, tanto desde el punto de vista técnico como de usuario. Esto incluyó la capacidad para realizar análisis de necesidades y establecer objetivos claros y alcanzables.

**Diseñar:** Se fortalecieron las habilidades para diseñar interfaces de usuario que no solo fueran visualmente atractivas, sino también funcionales y fáciles de usar. Esto incluyó la selección y aplicación de componentes de UI como los ofrecidos por PrimeReact (*PrimeReact | React UI Component Library*, s. f.-b). Se adquirieron habilidades en la arquitectura y modelado de sistemas, asegurando que el diseño técnico fuera escalable y eficiente (programación modular).

**Implementar:** Se mejoraron las habilidades de programación en React (*React*, s. f.), incluyendo la integración de bibliotecas externas como DataTables (*DataTables | JavaScript Table Library*, s. f.) y Axios (*Axios*, s. f.). La experiencia adquirida en la implementación de módulos y funcionalidades específicas fue crucial.

**Operar:** Se aprendió a realizar tareas de mantenimiento y soporte post implementación, asegurando que las funcionalidades se mantuvieran operativas



y eficientes a lo largo del tiempo. De igual manera se desarrolló la habilidad para identificar áreas de mejora y aplicar cambios que optimizaran la performance y usabilidad del sistema, basándose en el feedback de los usuarios.

## **Capítulo 6. Conclusiones**

El desarrollo de habilidades CDIO se ha visto permanentemente durante todo el periodo de práctica, el proyecto ha permitido aplicar y desarrollar estas habilidades, Cada una de estas ha sido esencial para el éxito del proyecto, desde la definición de los requisitos hasta la implementación efectiva y el mantenimiento del sistema.

Los objetivos planteados al inicio del proyecto han sido alcanzados en su totalidad. La implementación de los módulos clave y la creación de una interfaz de usuario fluida y responsiva han demostrado la efectividad de la planificación y el diseño. Aunque se enfrentaron desafíos, como la integración de múltiples tecnologías, se lograron cumplir los objetivos establecidos.

### **6.1 Trabajo Futuro**

**Expansión de Funcionalidades:** Considerar la adición de nuevos módulos o nuevas funcionalidades basadas en las necesidades emergentes de los usuarios, como características avanzadas de campos de entradas dinámicos.

**Integración de Tecnologías Emergentes:** Explorar la integración de tecnologías emergentes que puedan añadir valor al sistema.

### **6.2 Reflexión Final**

El desarrollo de este proyecto ha proporcionado una valiosa experiencia en habilidades como concebir, diseñar, implementar y operar, La programación con React (*React*, s. f.) y la integración de bibliotecas externas, fortalecieron habilidades en diseño de interfaces, resolución de problemas y gestión de proyectos. Las lecciones aprendidas durante el proyecto, como la importancia de la planificación detallada y la adaptabilidad a cambios, serán fundamentales para futuros proyectos. La experiencia adquirida ha sido crucial para el crecimiento profesional y la mejora continua en el campo del desarrollo de software.



## Referencias

1. [SIBUCSC - UCSC. \(s. f.\).](#)
2. [PrimeReact | React UI Component Library. \(s. f.\).](#)
3. [React. \(s. f.\).](#)
4. [JavaScript | MDN. \(2023, 24 julio\). MDN Web Docs.](#)
5. [DataTables | Javascript table library. \(s. f.\).](#)
6. [SweetAlert2. \(s. f.\). A Beautiful, Responsive, Customizable And Accessible \(WAI-ARIA\) Replacement For JavaScript's Popup Boxes.](#)
7. [Axios. \(s. f.\).](#)
8. [ReactDOM – React. \(s. f.\). React.](#)
9. [Contributors, M. o. J. T. A. B. \(s. f.\). \*Bootstrap\*.](#)



**UCSC**

**UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN  
VICERRECTORÍA ACADÉMICA  
DIRECCIÓN DE DOCENCIA  
FACULTAD DE INGENIERÍA**