

# **ESCUELA POLITÉCNICA NACIONAL**

**ESCUELA DE FORMACIÓN DE TECNÓLOGOS**

**DESARROLLO DEL SISTEMA WEB DE GESTIÓN DE  
CONTRASEÑAS**

**DESARROLLO DE UN SISTEMA WEB**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR  
EN DESARROLLO DE SOFTWARE**

**VALENZUELA SANDOVAL MICHAELLE AURELIO**

**DIRECTOR: ING. LORENA ELIZABETH CHULDE OBANDO**

**DMQ, marzo 2024**

## **CERTIFICACIONES**

Yo, **MICHAELLE VALENZUELA** declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

**Michaelle Aurelio Valenzuela Sandoval**

**michaelle.valenzuela@epn.edu.ec**  
**michaelle1.1valenzuela@gmail.com**

Certifico que el presente trabajo de integración curricular fue desarrollado por **MICHAELLE VALENZUELA**, bajo mi supervisión.

---

**LORENA CHULDE**  
**DIRECTOR**

**lorena.chulde@epn.edu.ec**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

**Michaelle Aurelio Valenzuela Sandoval**

## **DEDICATORIA**

Este trabajo y todo lo que hay detrás lo dedico a mi madre María Valenzuela, la cual es el pilar fundamental de mi vida, la persona que me da aliento en cada bache y problema que se presentó a lo largo de todo este proceso.

A mi familia como mis abuelitos, tíos, primos, amigos y demás personas que me alentaron a seguir.

**Michaelle Aurelio Valenzuela Sandoval**

## **AGRADECIMIENTO**

Agradezco en primer lugar a Dios, por escuchar cada una de las suplicas y lamentos que encomendé, por escuchar a mi madre y ser nuestro mayor consuelo.

Agradezco a mi madre por todo, absolutamente todo su esfuerzo y cariño, sin todo ese apoyo este proyecto no fuera posible, gracias no me alcanza la vida para devolverte todo lo que has hecho por mí.

Gracias a la Escuela Politécnica Nacional por permitirme estudiar esta hermosa carrera y completarla. A cada uno de los docentes que con su conocimiento y guía me acompañaron en cada paso. Agradezco de manera especial a mi tutora la Ing. Lorena Chulde por ayudarme a completar este trabajo y guiarme en todo lo que conllevó este proyecto. Finalmente agradezco a todos los que apoyaron a completar este proyecto y todo mi paso por la carrera.

Muchas gracias por todo.

**Michaelle Aurelio Valenzuela Sandoval**

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN .....	VII
ABSTRACT .....	VIII
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO .....	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos .....	2
1.3 Alcance .....	2
1.4 Marco Teórico .....	3
2 METODOLOGÍA .....	6
2.1 Metodología de Desarrollo .....	6
Roles.....	7
Artefactos.....	8
Recopilación de Requerimientos .....	8
Historias de Usuario.....	9
Product Backlog.....	10
Sprint Backlog.....	11
2.2 Diseño de interfaces .....	12
Herramienta utilizada para el diseño .....	13
2.3 Diseño de la arquitectura .....	13
Patrón arquitectónico .....	14
2.4 Herramientas de desarrollo.....	15
3 RESULTADOS.....	16
3.1 <i>Sprint</i> 0. Configuración del ambiente de desarrollo, Módulo de inicio de sesión y Módulo de administrador.....	16
Descripción del requerimiento y limitaciones para el proyecto .....	16
Estructura del proyecto.....	19
Roles de usuario.....	21
Módulo de inicio y cierre de sesión.....	21

Módulo de administrador: Administración de cuentas .....	22
Módulo de administrador: Seleccionar el tipo de encriptación. ....	23
3.2 <i>Sprint 1</i> Resultados del módulo de Administrador: Gestión de los tipos de recursos, módulo de registro y modificar perfil.....	25
Módulo de administrador: Gestionar los tipos de recurso que puede guardar el usuario. ....	25
Módulo de Registro.....	26
Módulo de modificar perfil.....	28
3.3 <i>Sprint 2</i> Resultados de los módulos de usuario: creación de un recurso, generador de contraseñas y contraseñas robustas. ....	30
Crear una nueva entrada del recurso a autenticar.....	30
Generar una nueva contraseña segura .....	32
Modificar los parámetros necesarios para generar una contraseña segura ..	33
3.4 <i>Sprint 3</i> Resultados de los módulos de usuario: modificación de recursos almacenados, visualización de información y selección de tipos de recursos. ...	34
Modificar los recursos almacenados.....	35
Visualizar la información sensible de los recursos previos .....	36
Seleccionar el tipo de recursos que desea guardar .....	37
3.5 <i>Sprint 4</i> Testeo del sistema <i>web</i> y despliegue.....	38
Pruebas unitarias y resultados.....	39
Pruebas de carga y estrés .....	40
Pruebas funcionales .....	41
Despliegue del sistema <i>web</i> .....	42
4 Conclusiones .....	44
5 Recomendaciones .....	45
6 REFERENCIAS BIBLIOGRÁFICAS.....	46
7 ANEXOS.....	51
ANEXO I .....	52
ANEXO II .....	53
ANEXO III .....	105
ANEXO IV.....	106

## RESUMEN

El uso de diversas plataformas ya sean de entretenimiento, entidades financieras, redes sociales y demás servicios disponibles en la red se acceden con credenciales de autenticación: un usuario o correo y una contraseña. El número de credenciales a recordar y guardar se complica conforme pasa el tiempo, haciendo más difícil al usuario poder recordar en especial las contraseñas. Por lo cual opta por usar la misma contraseña en todas las plataformas o servicios y además con contraseñas muy cortas y fáciles de vulnerar.

Para solucionar estos fallos de seguridad que comprometen información sensible del usuario, se plantea el desarrollo del sistema *web* de Gestión de contraseñas el cual está conformado por módulos como el registro de recursos los cuales son: credenciales de autenticación, tarjetas de crédito y notas seguras. Siendo estos tres los recursos principales. Por su parte el Administrador del sistema puede añadir más recursos que pueden ser útiles para el usuario abarcando un mayor número de información sensible.

El desarrollo del sistema se basa en la metodología *Scrum*, para lo cual el trabajo de integración curricular es dividido en secciones. La primera sección se detalla el componente de desarrollo, los objetivos, el alcance del sistema y el marco teórico. La segunda sección se describe la aplicación de la metodología *Scrum*, el prototipado y las herramientas usadas en el sistema. La tercera sección muestra los resultados del desarrollo del sistema dividido en *Sprints*. Como penúltima sección se detalla las conclusiones obtenidas a lo largo del desarrollo y como último punto las recomendaciones.

**PALABRAS CLAVE:** Sistema *web*, contraseña, seguridad, plataforma, credenciales.



## **ABSTRACT**

The use of various platforms, whether entertainment, financial entities, social networks and other services available on the Internet, is accessed with authentication credentials: a username or email and a password. The number of credentials to remember and save becomes more complicated as time goes by, making it more difficult for the user to remember passwords, especially. Therefore, choose to use the same password on all platforms or services and also with very short passwords that are easy to break.

To solve these security flaws that compromise sensitive user information, the development of the password management web system is proposed, which is made up of modules such as the registration of resources, which are: authentication credentials, credit cards and secure notes. These three resources being the main ones. For his part, the System Administrator can add more resources that may be useful to the user, covering a greater number of sensitive information.

The development of the system is based on the Scrum methodology, for which the curricular integration work is divided into sections. The first section details the development component, the objectives, the scope of the system and the theoretical framework. The second section describes the application of the Scrum methodology, prototyping and the tools used in the system. The third section shows the results of the development of the system divided into Sprints. The penultimate section details the conclusions obtained throughout the development and as the last point the recommendations.

**KEYWORDS:** Web system, password, security, platform, credentials.

# 1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

En la actualidad el número de usuarios que hacen uso de muchos sistemas informáticos de la *web* y en la mayoría de estos sitios piden el ingreso de las credenciales de autenticación para acceder a los diferentes servicios. Según el Incibe hasta el 2019 aproximadamente el 90% de las contraseñas usados en estos sitios son vulnerables, esto basado en diferentes sondeos realizados a los usuarios en donde afirman que la contraseña más utilizada es el “123456” [1].

En muchos casos ingresan contraseñas cortas, débiles o intuitivas para evitar el olvido de estas, siendo fáciles de vulnerar y representando un peligro para el usuario. Se hacia especial mención al cambio periódico y continuo de las contraseñas de manera manual pues que se creía que aumentaban el nivel de seguridad, en los últimos tiempos se está llegando al conceso de que esto puede ser contraproducente puesto que hacerlo de manera personal los usuarios tienden a seguir patrones en cada contraseña modificada haciendo que estén susceptibles a ser descifradas en cadena siguiendo el mismo patrón [2].

En Latinoamérica el asunto con respecto a la seguridad informática es muy grave, puesto que se estima que esta región sufrió un aproximado de 360 millones de ataques informáticos en 2022 [3]. Para implementar seguridad en las credenciales se pueden usar caracteres alfanuméricos, letras mayúsculas y minúsculas e incluso caracteres especiales. Según el estudio publicado por Hiven Systems los ataques de fuerza bruta los cuales son utilizados para la obtención de credenciales, se basan en el poder de procesamiento del ordenador y de los últimos de la unidad grafica dando como resultado que una contraseña que solo se base en números puede ser descifrada de manera inmediata, al contrario de una contraseña que tiene caracteres alfanuméricos y especiales el tiempo se dispara a más de 3 mil años [4].

Al generar credenciales de autenticación en muchos sistemas las personas tienden a olvidarse, por lo cual en una gran mayoría de situaciones guardan la información en archivos planos como de ofimática o de manera física como en papeles o notas adhesivas, siendo fácil de acceder, modificar o sustraer. Por otra parte, los usuarios

tienden a usar una única contraseña para varios sistemas creando una brecha de seguridad en la autenticación.

Para facilitar la gestión de un número considerable de contraseñas, usuarios y demás información sensible se propone el desarrollo del SISTEMA WEB DE GESTIÓN DE CONTRASEÑAS, dando paso a la simplificación del almacenamiento y la generación de contraseñas seguras, principio fundamental de la seguridad informática.

### **1.1 Objetivo general**

Desarrollar un sistema web de gestión de contraseñas mediante los *frameworks* VueJs y Express para el *frontend* y el *backend* respectivamente para resguardar la información sensible de los usuarios.

### **1.2 Objetivos específicos**

1. OE1 Desarrollar el *backend* del sistema en base a los requerimientos para la generación de las *APIrest* y levantar la base de datos.
2. OE2 Desarrollar el *frontend* del sistema de forma sencilla, amigable y fácil de usar mediante el uso de las buenas prácticas del diseño UX/UI para que el usuario navegue de forma rápida e intuitiva.
3. OE3 Desplegar el sistema a producción utilizando un servicio de hosting de Vercel y Koyeb para la disponibilidad de los usuarios.

### **1.3 Alcance**

El siguiente Trabajo de Integración Curricular (TIC) plantea el despliegue de un sistema web para la gestión de contraseñas de tal manera que el usuario tenga fácil acceso a los diferentes registros de manera, segura, rápida y en tiempo real. Se considera los siguientes perfiles:

Usuario: podrá hacer uso del sistema de modo que pueda guardar nuevas entradas con información como: contraseñas, usuarios, correos electrónicos, además de

generar contraseñas en base a las necesidades del usuario o del sitio en donde desea utilizar dicha contraseña. De igual forma podrá modificar o eliminar la información guardada con anterioridad.

**Administrador:** encargado de modificar diferentes aspectos críticos del sistema como la gestión de usuario, la gestión de campos de registro o realizar modificación en el apartado de seguridad.

Para controlar el cumplimiento de los tiempos establecidos, el desarrollo de las funcionalidades, la codificación de los módulos que cada perfil es necesario emplear de una metodología ágil: *Scrum*. Como parte del marco de trabajo de *Scrum* es necesario la definición de roles y tiempos de entrega de los módulos o funcionalidades en base a los requerimientos recopilados del usuario. En cumplimiento con las etapas de desarrollo, una vez terminada la etapa de codificación como parte del control de calidad del producto se realiza pruebas que demuestran el correcto funcionamiento del módulo acorde a cada perfil y la totalidad del sistema.

Para cada una de las etapas del proyecto se emplean herramientas de desarrollo que ayudan en el cumplimiento de los requerimientos, como herramientas de diseño que permiten dar un primer vistazo del proyecto en cada una de sus funcionalidades, editores de texto que permiten una mejor productividad por parte del equipo de desarrollo. La etapa de pruebas emplea herramientas que garantizan la calidad del proyecto y como parte final el despliegue a producción para dar por finalizado el desarrollo del sistema.

## 1.4 Marco Teórico

**Backend:** o lado del servidor, constituye la parte a cargo de la gestión y procesamiento de datos, incluyendo la lógica de la aplicación, la seguridad y autenticación de los usuarios. Es responsable de procesar las solicitudes, traer la información del banco de datos y de enviarla al *frontend* [5].

**API RESTful:** es un estilo arquitectónico empleado para el diseño de servicios *web* que utiliza los principios REST. Implementa los métodos HTTP estándar y las operaciones CRUD a los recursos. Esta arquitectura facilita la comunicación entre sistemas de manera eficiente y escalable [6].

**Base de datos:** definido como un sistema organizado de datos que se almacenan y gestionan de tal manera que se pueda recuperar, acceder y administrar fácilmente. Se dividen en relacionales y no relacionales, donde la principal diferencia radica en la organización de los datos y en cómo están comunicados unos con otros [7].

**El *frontend*:** se refiere al segmento de un sitio o sistema *web* responsable de la interacción del usuario con el sistema. El *frontend* engloba el maquetado y diseño de la interfaz de usuario, dando como resultado la parte visual completa del producto como también la interacción con el *backend* [8].

***Framework*:** es un conjunto estructurado de herramientas, bibliotecas, librerías y estándares empleado en el desarrollo de *software*. Tiene el objetivo de facilitar y estandarizar la creación y organización de un sistema o aplicación. Mejora de manera significativa la construcción y el orden del código facilitando la reutilización y la solución de posibles problemas [9].

**Un cliente HTTP:** *software* dedicado al envío y recepción de solicitudes y respuestas HTTP. Permite obtener, enviar o manipular datos de la *web*. Útil en el consumo de APIs y el testeo de *endpoints* en conjunto con las solicitudes de GET, POST, DELETE, etc [10].

**API:** conjunto de reglas y definiciones que posibilita el intercambio de información entre diferentes sistemas de *software*. Proporciona una puerta de comunicación que permite el uso de funcionalidades y datos de un sistema en otro sin alterar la lógica interna de ninguno [11].

**El diseño *web responsive*:** hace referencia a la práctica de crear sitios que ofrecen una experiencia óptima de usuario en una variedad de dispositivos y tamaños de pantalla. La creación de sitios responsivos evita la implementación de múltiples versiones para diferentes dispositivos, en base a que el sitio responde y se adapta de manera automática al dispositivo que se está usando [12].

**UI:** significa Interfaz de usuario, se refiere al espacio de interacción entre el usuario y el sistema. La UI abarca todos los elementos visuales como botones, menús, formularios, gráficos y demás elementos de interacción que dan paso a la

comunicación del usuario con el sistema de forma efectiva y comprensible para ambos actores [13].

**UX:** experiencia de usuario del inglés *User Experience* se refiere a la experiencia general que tiene un usuario al momento de interactuar con un producto o servicio. Los aspectos más importantes en la experiencia de usuario son la usabilidad, la accesibilidad, el diseño, la fluidez de interacción y la satisfacción general al momento de usar el producto [14].

**Métodos de encriptación:** son técnicas utilizadas para intervenir información legible a un formato el cual es ilegible, denominado cifrado, con la finalidad de proteger la privacidad de los datos. Estos métodos usan algoritmos matemáticos que emplean claves para cifrar y descifrar los datos [15].

**El hashing:** es un proceso matemático que convierte cierta cantidad de datos en una cadena de longitud fija. El resultado de este proceso se define como “*hash*” o “valor *hash*”. Se emplean con regularidad para garantizar la integridad de los datos [16].

**Las pruebas unitarias:** son una práctica que se emplea para evaluar de forma independiente porciones aisladas de código, como una clase o método. El propósito es garantizar que cada parte del *software* responda como está previsto y evitar errores de manera temprana [17].

**Las pruebas de carga:** son un subconjunto de pruebas que evalúan el rendimiento y la capacidad de un sistema o sitio al ser sometido a una carga significativa. El objetivo es entender cómo se comporta el sistema al exponerse a una gran cantidad de usuarios, transacciones y solicitudes. Los resultados identifican posibles cuellos de botella, problemas de rendimiento y la escalabilidad del proyecto [18].

**Las pruebas de rendimiento:** son evaluaciones que se usan para medir y verificar la capacidad, velocidad y estabilidad de un sistema en respuesta a una carga específica. Estas evaluaciones son necesarias para identificar diferentes problemas de rendimiento y mejorar los tiempos de carga, en general toda la experiencia del usuario [19].

## 2 METODOLOGÍA

Se toma como punto de partida para el desarrollo de este proyecto el uso de metodologías de aprendizaje como es el estudio de casos de uso, el cual es aplicable y replicable para varias áreas de conocimiento. El estudio de casos de uso describe de forma detallada y precisa un caso particular con el fin de identificar el funcionamiento individual de sus partes, los problemas a los cuales este sujeto y las diferentes soluciones aplicables [20].

### 2.1 Metodología de Desarrollo

Son marcos de trabajo que tienen por objetivo determinar los procesos y tareas de las etapas de planificación, diseño, desarrollo y entrega de un producto de *software*. La metodología determina los pasos, actividades y roles fundamentales para trabajar los proyectos de la manera más óptima y productiva posible. Entre algunos de los beneficios que aporta al equipo y el proyecto está la calidad del producto, incentivar el trabajo en equipo y la participación de todos los actores a lo largo de todo el proceso de creación del sistema [21].

Por su parte una metodología ágil es una perspectiva de trabajo participativo y versátil diseñado para la construcción de productos de *software* que tiene como eje principal la capacidad de cambio y la entrega gradual. Se fundamenta en la iteración y la evaluación constante provista por parte del equipo de desarrollo como del usuario, lo cual da como resulta soluciones tempranas a los cambios planteados y maximizar el desarrollo de código significativo para el proyecto [22].

La metodología *Scrum* es la seleccionada para ayudar en el cumplimiento de los requerimientos y tiempos de desarrollo y entrega que se han definido al inicio de este proyecto. Esta metodología es óptima y adecuada para proyectos complejos en donde los obstáculos están presentes en el entorno de desarrollo, donde la característica principal es la obtención de resultados breves y entregas continuas. *Scrum* se basa en la entrega de partes individuales de *software* que son importantes para el sistema que den valor real al avance del proyecto, estos resultados se dividen en ciclos cortos llamados iteraciones o *Sprints*. Como parte de la

metodología *Scrum* es necesario definir puntos importantes sobre los cuales está basado este Trabajo de Integración Curricular.

### **Roles**

Los roles son puesto importantes entre el equipo de trabajo que ejercen diferentes actividades concretas para alcanzar los tiempos y objetivos del proyecto. Cada uno de estos actores tiene ciertas funciones y obligaciones acorde al puesto que ocupan [23]. En base a esta definición se describen los siguientes roles los cuales forman parte del proyecto y sus resultados.

#### ***Product Owner***

El responsable de defender las necesidades del cliente al igual que es encargado de administrar los puntos del *Producto Backlog* y de comunicar el punto de vista del sistema en función del modelo del negocio [24]. Acorde a esta definición en la **Tabla** se muestra al responsable de cumplir con este rol.

#### ***Scrum Master***

Es la persona que agiliza el avance de la metodología y de asegurar que se usan de manera conjunta todas la buenas prácticas y principios de *Scrum*. Trabaja para eliminar los obstáculos, promover la aplicación de los principios y prácticas de *Scrum*, fomenta la colaboración y autogestión del equipo[25]. En la **Tabla** se presenta al responsable de desempeñar estas actividades.

#### ***Development Team***

Es el grupo encargado de realizar las entregas periódicas de las funcionalidades al final de cada *Sprint*. El equipo debe ser capaz de abordar cada las diferentes eventualidades que conlleve el desarrollo del sistema y cumplir con los requerimientos y cambios propuestos por el usuario [26]. En la **Tabla** se muestra al responsable de llevar a cabo estas actividades.



**Tabla 2.1. Responsables de todos los roles.**

NOMBRE	ROL
Ing. Lorena Chulde	<i>Product Owner</i>
Ing. Lorena Chulde	<i>Scrum Master</i>
Michaelle Valenzuela	<i>Development Team</i>

### **Artefactos**

Los artefactos son recursos utilizados para corroborar la implementación del sistema y guardar los progresos del desarrollo. Estos recursos entregan información que ayuda a mantener la dirección del proyecto clara y verificar la productividad del *Sprint* y como se están gestionan los recursos empleados[27]. Los artefactos detallados a continuación son: Recopilación de requerimientos, *Product Backlog* y *Sprint Backlog*.

#### **Recopilación de Requerimientos**

Artefacto por el cual se reconoce, registra y asimila las necesidades y perspectivas de todo el sistema. Se identifica por iniciar la planificación y dar paso a las primeras etapas del proyecto[28]. Con los requerimientos levantados se registra la siguiente **Tabla 2.2**, el resto del contenido de la Recopilación de requerimientos se encuentra en el **ANEXO II**.

**Tabla 2.2 Descripción de los requerimientos del usuario.**

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
Sistema web	RR002	<p>El administrador necesita consumir varios <i>endpoints</i> para:</p> <ul style="list-style-type: none"> <li>• Cambiar el estado de la cuenta de los usuarios (activo a inactivo).</li> </ul>

### Historias de Usuario

Es un artefacto el cual describe de manera breve y centrada una funcionalidad en base a las necesidades del usuario. Es usado como marco de trabajo en la etapa de planificación de un Sprint. Las historias de usuario se redactan desde la perspectiva del usuario y se centra en los resultados que se espera. Las historias de usuarios deben ser escritas de manera simple y con la mayor cantidad de detalles como: el perfil objetivo, la prioridad y riesgo en desarrollo y en el negocio, una descripción, el alcance de la funcionalidad a desarrollarse, observaciones al momento de desarrollo y los criterios de aceptación [29]. En la **Tabla 2.3** se muestra una historia correspondiente a este proyecto, el resto de las historias recopiladas se pueden encontrar en el **ANEXO II**.

**Tabla 2.3. Demostración de Historia de usuario.**

HISTORIA DE USUARIO	
ID: HU-007	Usuario: Usuario
Nombre de la Historia: Modificar el perfil de usuario	
Prioridad para el negocio: Alto	Riesgo para desarrollo: Bajo
Desarrollador responsable: Michaelle Valenzuela	
<p><b>Descripción:</b>  <b>Como:</b> Usuario  <b>Quiero:</b> Modificar el perfil de usuario  <b>Para:</b> Para actualizar la información ingresada durante el registro</p>	

<p><b>Alcance:</b></p> <ol style="list-style-type: none"> <li>1. El botón “Modificar perfil” dentro de los recursos del usuario abrirá una ruta con los siguientes campos a rellenar: <ol style="list-style-type: none"> <li>a. Correo</li> <li>b. Contraseña</li> <li>c. Repita la nueva contraseña</li> </ol> </li> <li>2. Botón “Guardar” que permitirá guardar la información del usuario a ser modificada.</li> <li>3. Botón “Cancelar” que permitirá deshacer (cerrar la página) de modificar el perfil de usuario.</li> </ol>
<p><b>Observación:</b></p> <ul style="list-style-type: none"> <li>• <b>Correo</b> (campo de texto): debe cumplir con el formato de correo valido.</li> <li>• <b>Contraseña</b> (campo de texto): debe tener un mínimo de 8 caracteres y no debe ser igual a la contraseña ya guardada.</li> <li>• <b>Repita la nueva contraseña</b> (campo de texto): repetir la contraseña ingresada en el campo de “Contraseña”, esta debe ser la misma para ser valida.</li> <li>• <b>Botón “Guardar”</b>: que actualizara de manera inmediata la nueva información en la BDD.</li> <li>• <b>Botón “Cancelar”</b>: que descartara la información ingresada por el usuario y cerra la página de modificar el perfil de usuario.</li> </ul>
<p><b>Criterios de Aceptación:</b></p> <ul style="list-style-type: none"> <li>• Se presentará un mensaje de error cuando la actualización no se realizó de manera satisfactoria, junto con el porque de la no ejecución de dicha acción.</li> <li>• Se presentará un mensaje a modo de alerta cuando la información se actualizo de forma satisfactoria.</li> </ul>

### Product Backlog

Artefacto presentado como una lista priorizada con todas las funcionalidades, actividades, y cambios a realizarse a lo largo de todo el proyecto. Esta lista es trabajada por parte del *Product Owner* y *Scrum Master*, responsables de realizar los cambios pertinentes de acuerdo a las necesidades del negocio o el cliente [30]. En la **Tabla 2.4** Muestra del *Product Backlog*., se muestra una parte de la lista del artefacto, el resto de las funcionalidades se encuentran en el **ANEXO II**.

**Tabla 2.4 Muestra del *Product Backlog*.**

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID – HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU007	Modificar el perfil de usuario	3	Finalizada	Alta
HU008	Generar una nueva contraseña segura	3	Finalizada	Alta

### **Sprint Backlog**

Artefacto en forma de lista redactado de forma específica basada en los elementos del *Product Backlog* que se tiene que realizar a lo largo de todo el *Sprint*. Detalla todas las tareas y actividades importantes para convertir las funcionalidades individuales en entregables para el usuario [31]. En base a esta definición se presenta en la **Tabla 2.5** una muestra parcial de las tareas de un *Sprint*, el resto de la información se encuentra en el **ANEXO II**.

**Tabla 2.5 Muestra del *Sprint Backlog*.**

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>						
SB 003	Diseño e implementación del perfil usuario.	Módulo – Nuevo recurso	HU006	Crear una nueva entrada del recurso a autenticar	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para crear una nueva entrada con información a almacenar.</li> <li>• Validación de los campos requeridos.</li> </ul>	40 H

		Módulo – Modificar el perfil	HU007	Modificar el perfil de usuario	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para modificar la información proporcionada al momento de registrarse.</li> <li>• Validación de los campos requeridos.</li> </ul>
		Módulo - Generar nueva contraseña	HU008	Generar una nueva contraseña segura	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para generar una contraseña predeterminada segura</li> <li>• Validar los campos requeridos.</li> </ul>

## 2.2 Diseño de interfaces

La etapa de diseño está destinada a la creación visual y las acciones que el usuario puede tener con el sistema o sitio. Entre los elementos que pueden ser creados, modificados o retirados esta los recursos visuales, gama de colores, tipografía y la experiencia de usuario. Toda esta creación debes estar basada en los requerimientos previos y los cambios propuestos por el usuario [33]. Para agilizar la creación de la experiencia de usuario y todo lo que esto conlleva es necesario el uso de herramientas destinadas para este propósito.

## Herramienta utilizada para el diseño

La herramienta seleccionada para facilitar el desarrollo de la etapa de diseño es Figma. Entre muchas de sus características positivas está el uso de *plugins*, componentes y la posibilidad de generar la simulación de un entorno de producción del sistema, realizar el trabajo desde la nube y la participación colaborativa en tiempo real [32]. En la **Figura 2.1** se muestra el resultado del maquetado de una de las ventanas del sistema, los demás resultados se encuentran en el **ANEXO II**.



**Figura 2.1** Muestra del prototipado del sistema.

## 2.3 Diseño de la arquitectura

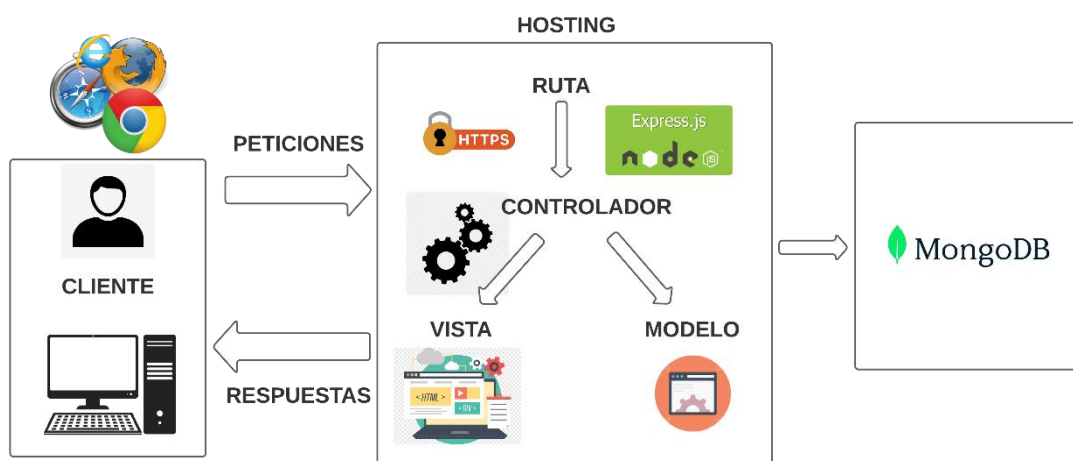
Se refiere al proceso de organización y estructuración de un sistema o proyecto para asegurar que se cumplan con los requisitos y metas sin descuidar la eficiencia y la sostenibilidad. El diseño y empleo de una arquitectura adecuada al proyecto garantiza que este sea escalable y mantenible a lo largo del tiempo[32]. Parte del diseño de la arquitectura del sistema es la elección de un patrón arquitectónico el cual es definido en el siguiente apartado.

## Patrón arquitectónico

Se describe como una solución general y reutilizable para abordar un problema que es considerado repetitivo en el desarrollo de *software*. El patrón arquitectónico ofrece guías y practicas útiles de cómo resolver y desarrollar la relación entre los diferentes componentes que forman parte de un sistema. El patrón seleccionado para este proyecto tiene tres componentes: Modelo, Vista y Controlador [33]. Cada una de estas capas es definida de la siguiente manera:

- **Modelo:** es la responsable del manejo y uso de datos al igual que de la lógica del funcionamiento del sistema.
- **Vista:** a cargo de reflejar la interfaz de usuario y los de datos de manera visual.
- **Controlador:** controla las acciones del usuario con el sistema proporcionado un puente de comunicación entre el modelo y la vista.

Como muestra del patrón seleccionado en la **Figura 2.2** se presenta los componentes que interaccionan entre ellos.



**Figura 2.2 Patrón de arquitectura.**

## 2.4 Herramientas de desarrollo

Responsables de facilitar el proceso de creación de un sistema de *software*. Las herramientas mejoran el flujo del trabajo y simplifican de manera significativa los tiempos de codificación y la posible aparición de errores[34]. Para muestra en la **Tabla 2.6** se encuentra algunas de las herramientas empleadas para este sistema.

**Tabla 2.6:** Muestra de las herramientas de desarrollo.

HERRAMIENTA	JUSTIFICACIÓN
<p><b>Vue.js</b></p> 	<p><i>Framework</i> empleado para el desarrollo de IU interactivas con curva de aprendizaje suave y basado en componentes [35].</p>
<p><b>Vue CLI</b></p> 	<p>Entorno de ejecución de comandos que facilita la gestión y creación de proyectos de Vue.js al igual que la integración de complementos como pruebas y empaquetado [36].</p>
<p><b>Vue Router</b></p> 	<p>Biblioteca oficial de enrutamiento, navegación y sincronización de estados, rutas y componentes en aplicaciones Vue.js [37].</p>
<p><b>Postman</b></p>  <p>POSTMAN</p>	<p>Herramienta de <i>software</i> empleada para probar, desarrollar y documentar APIs. Su uso es fácil ya que implementa una interfaz de usuario gráfica, los usos más comunes son crear peticiones de puntos finales de la API, enviar varios tipos de datos y evaluar las respuestas obtenidas [38].</p>



### 3 RESULTADOS

Al llegar a esta sección, se presentan todas las soluciones generadas durante el proceso de desarrollo, asegurándose de que satisfacen los requerimientos levantados en etapas tempranas del proyecto y los futuros cambios impuestos por el usuario. Adicional a las evidencias del desarrollo de los módulos y funcionalidades se muestra el testeado de dichas porciones de código. Los resultados se presentan en *Sprints* que se apegan a los principios y practicas propias de la metodología *Scrum*.

#### 3.1 *Sprint 0*. Configuración del ambiente de desarrollo, Módulo de inicio de sesión y Módulo de administrador.

A lo largo del *Sprint* se debe completar las siguientes tareas de acuerdo con el *Sprint Backlog*:

- Definición de requerimientos y limitaciones del proyecto.
- Estructura del proyecto.
- Roles de usuario.
- Módulo de inicio de sesión
- Módulo de administrador: Administración de cuentas

#### Descripción del requerimiento y limitaciones para el proyecto

##### Iniciar y cerrar sesión

La página de inicio de sesión está disponible para el administrador como para el usuario, por defecto el administrador tendrá acceso con credenciales definidas por defecto. Si bien el usuario puede acceder al inicio de sesión solo podrá acceder al sistema si este se encuentra registrado.

### **Cambiar estado de la cuenta de los usuarios**

El rol de administrador le faculta la responsabilidad de gestionar las cuentas de los usuarios registrados. En el panel del administrador una vez iniciada la sesión se presentan diferentes opciones como la de activar y desactivar cuentas en el apartado de Administración de usuarios. El listado de usuarios que se recupera de la BDD de todo aquel usuario con cuenta en el sistema y verificado.

### **Seleccionar el tipo de encriptación datos**

En el panel de administración se presenta en la opción de seleccionar la encriptación de los datos de los usuarios dependiendo del recurso a usarse. El tipo de encriptación al ser seleccionado modifica el cómo se guarda los datos en la BDD de tal manera que esta puede ser más robusta al momento de guardarse.

### **Gestionar los tipos de recursos que puede guardar el usuario**

En el panel de administración se presenta el apartado de Administrar los recursos. En este apartado el administrador tiene la posibilidad de crear nuevos tipos de como el usuario puede guardar su información sensible. El nuevo tipo de recurso creado puede ser eliminado de ser necesario.

### **Registrarse**

El rol de usuario le permite completar un formulario con información personal la cual será guardada en la BDD, una vez terminado el registro el sistema notificará al correo ingresado que verifique su cuenta para tener completo acceso al sistema.

### **Crear una nueva entrada del recurso a autenticar**

El usuario con una cuenta registra y verificada tendrá la posibilidad de acceder al sistema y la creación de los diferentes tipos de recursos disponibles. En el apartado de creación de recurso se presenta un formulario con entradas correspondientes al tipo de recurso seleccionado.

### **Modificar el perfil de usuario**

El usuario con una cuenta activa puede modificar la información de su perfil de usuario de manera que pueda actualizar la información según los apartados disponibles.

### **Generar una nueva contraseña segura**

En el apartado de creación de un nuevo recurso el usuario accede a la generación de una contraseña segura. La contraseña presentada al usuario es una con los parámetros predefinidos, el uso de caracteres alfanuméricos y una longitud de 8 caracteres.

### **Modificar los parámetros necesarios para generar una nueva contraseña segura**

En el apartado de creación de un nuevo recurso el usuario puede extender y habilitar nuevas opciones para generar una nueva contraseñas más robusta o acorde a sus necesidades.

### **Modificar los recursos almacenados**

Los recursos creados son listados al momento de iniciar sesión por el usuario. Todos los recursos almacenados tienen la opción de ser actualizados con nueva información de ser el caso o de ser eliminados si el recurso ya no es necesario para el usuario.

### **Visualizar la información sensible de los recursos previos**

Los recursos listados al momento de iniciar sesión pueden ser visualizados por parte del usuario, lo que permite copiar aquellos campos que necesitan ser autenticados y cambiar la visibilidad de los campos ocultos por su grado de sensibilidad.

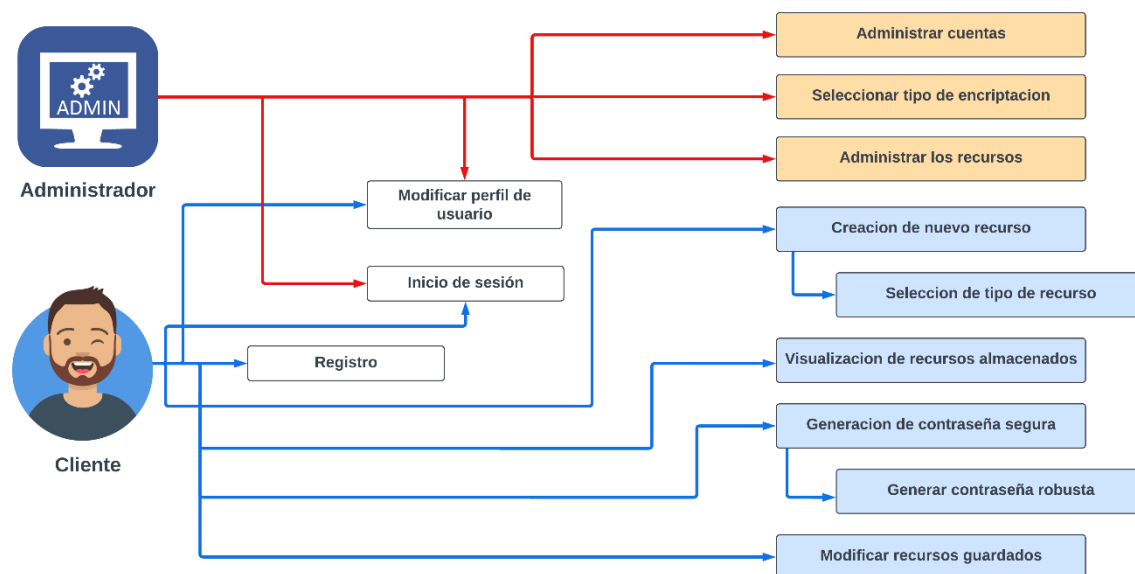
## Seleccionar el tipo de recurso que desea guardar

El usuario en el apartado de creación de un nuevo recurso puede seleccionar el tipo de recurso que se acomode a sus necesidades y que esté disponible para ser completado.

### Estructura del proyecto

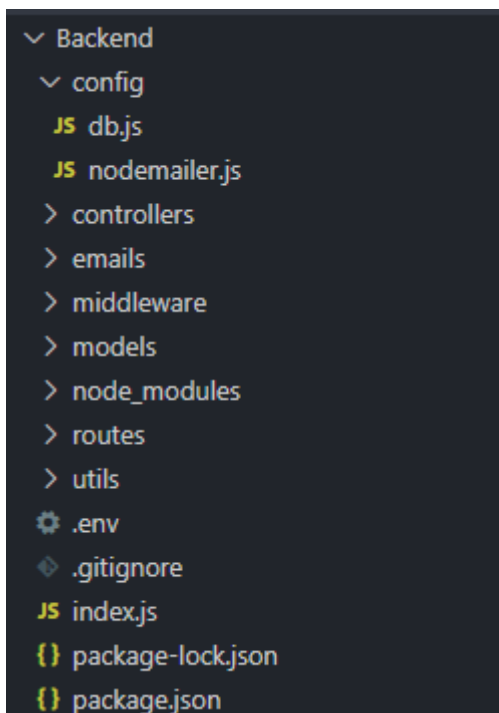
Como parte del planteamiento del proyecto se define el rol de administrador que se encuentra a cargo de los apartados de Administración de usuarios, Selección del tipo de encriptación de los datos y Administrar los recursos. Por parte del usuario se encuentra el uso del sistema en los apartados de creación, visualización y modificación de recursos, actualización de información del usuario y el registro en el sistema.

En la **Figura 3.1** se muestra todos los módulos a la que cada rol pueden acceder.

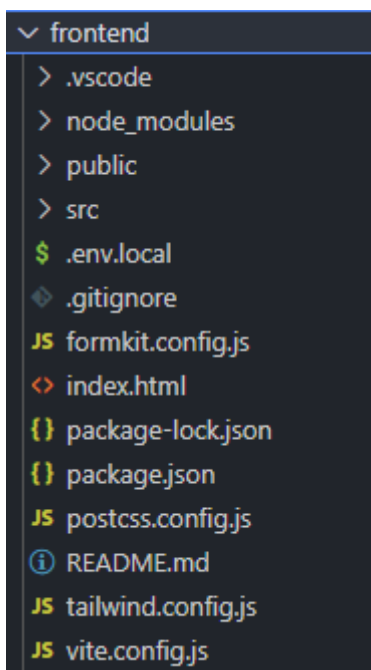


**Figura 3.1 Funcionalidades del administrador y usuario**

En la **Figura 3.2** y **Figura 3.3** se describe la estructura de carpetas empleada para el proyecto.

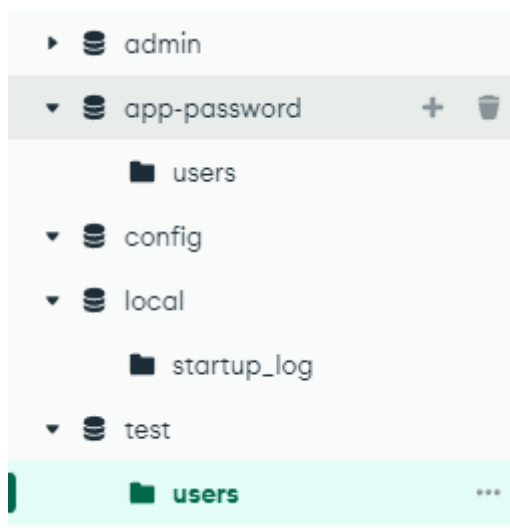


**Figura 3.2 Directorios del *Backend*.**



**Figura 3.3 Directorios del *Frontend*.**

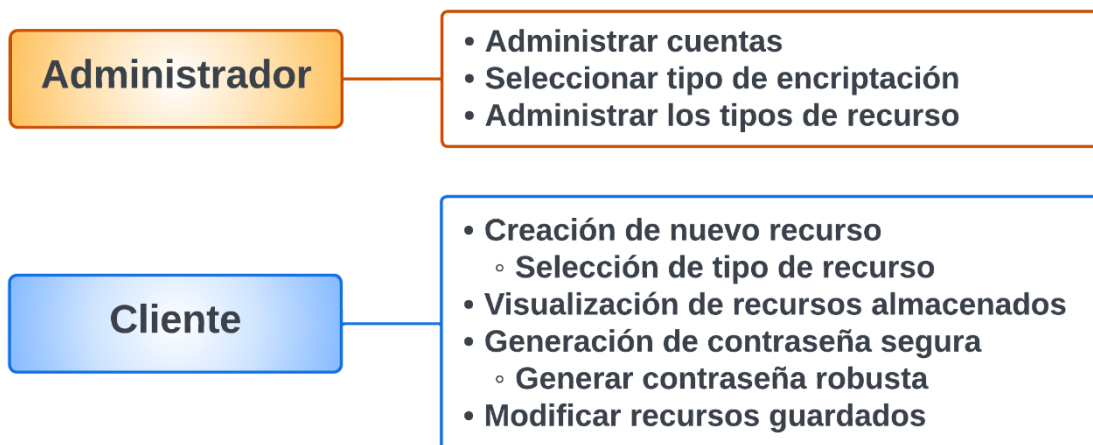
Se ha optado por emplear MongoDB para el almacenamiento de los datos, el cual permite la persistencia de datos de los usuarios, el administrador y los recursos. En la **Figura 3.4** se muestra la totalidad de las carpetas empleadas en el sistema.



**Figura 3.4 Directorios de MongoDB.**

### Roles de usuario

Como evidencia de los roles del sistema en la **Figura 3.5** se muestra las funcionalidades del administrador y usuario.

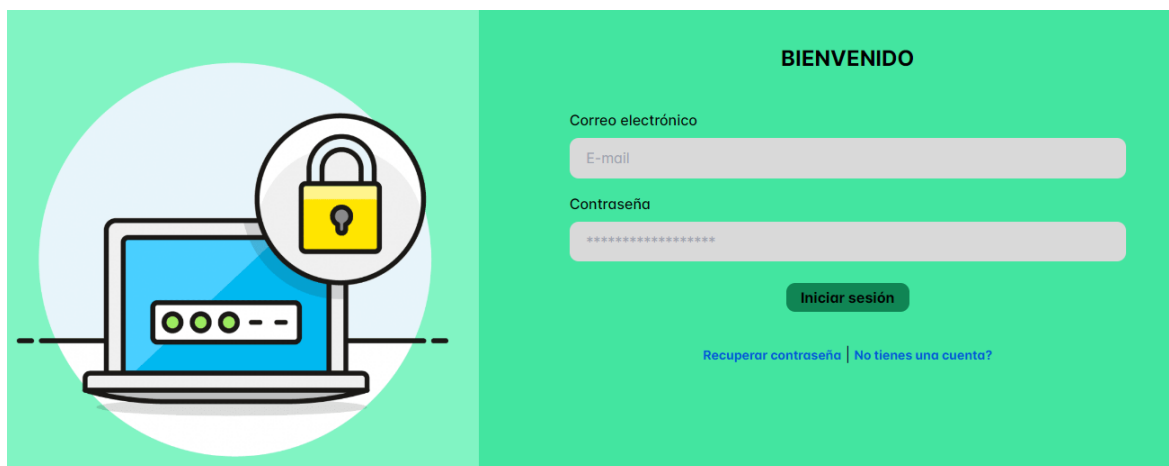


**Figura 3.5 Funcionalidades del sistema.**

### Módulo de inicio y cierre de sesión

El sistema web proporciona distintos *endpoints* e interfaces para el inicio y cierre de sesión. La lógica del sistema determina que para el inicio y cierre de sesión tanto el administrador como el usuario deben estar registrados. El usuario además debe

tener su cuenta verificada, si no cumple con esta directiva no podrá acceder al sistema. Con el módulo explicado, en la **Figura 3.6** se muestra el resultado obtenido durante el desarrollo en conjunto con la prueba unitaria que demuestra el correcto funcionamiento y su evidencia en la **Figura 3.7**.



**Figura 3.6** Página del módulo de inicio de sesión.



```
// Test de inicio de sesión de usuario
it("Inicio de sesion", async () => {
  const req: ReqLoginType = {
    body: {
      email: MailVerifier.targetUserEmail,
      password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await loginUser(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
  expect((res.json.mock.lastCall[0] as any).data.token_auth).toBeDefined();
  TOKEN = (res.json.mock.lastCall[0] as any).data.token_auth;
  USER_ID = decodeTokenUser(
    (res.json.mock.lastCall[0] as any).data.token_auth
  ) as JwpPayload;
}, 30000);
```

**Figura 3.7** Prueba unitaria del módulo de Inicio de sesión.

#### Módulo de administrador: Administración de cuentas

El sistema web cuenta con un apartado dedicado solo al Administrador del sistema el cual es responsable de diferentes funcionalidades como el de Administrar cuentas, para lo cual es sistema recupera todos los usuarios registrados y

verificados. Se enlista los usuarios y las cuentas pueden ser activadas o desactivadas a discreción del Administrador, dando como resultado el impedir que el usuario no pueda usar, acceder o recuperar su información. Determinada las funcionalidades del módulo, a continuación, se presentan las evidencias de la codificación en la **Figura 3.8** y la prueba unitaria en la **Figura 3.9**.

Nombre y Apellido	Estado	Acción
Nombre1 Apellido1	Inactivo	
Michaelle Valenzuela	Activo	

**Figura 3.8** Página del módulo de Administración de cuentas.

```
// Función que cambia el estado de un usuario
const changeUserState = async (userIsActive: boolean) => {
  const req: ReqResourceType = {
    body: {
      userIsActiveByAdmin: userIsActive
    },
    params: {
      id: TEST_USER_ID
    }
  };
  await deactivateUser(req, res);
  expect(res.status.mock.calls).toEqual([[201]]);
};
```

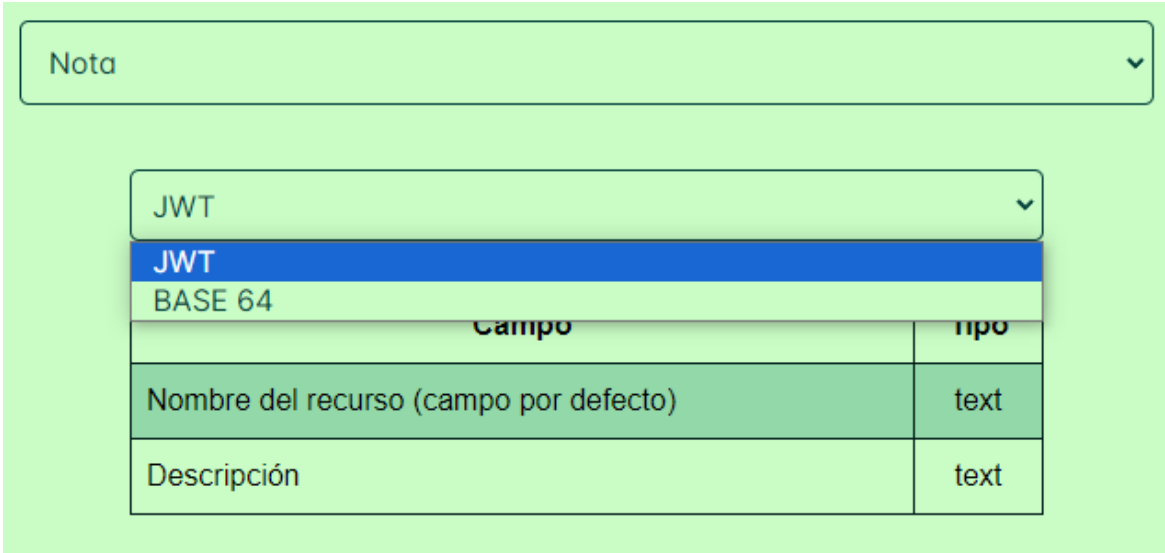
**Figura 3.9** Prueba unitaria del módulo de Administración de cuentas.

#### **Módulo de administrador: Seleccionar el tipo de encriptación.**

El panel de administración tiene la funcionalidad de cambiar las opciones de cómo se maneja la encriptación de los datos del usuario, las opciones están establecidas en la lógica del sistema. Al momento de cambiar de método de encriptación los datos que a posterior se guarden por el usuario en ese recurso se encriptaran de diferente en al BDD. Como evidencia del desarrollo realizado en este módulo en la



**Figura 3.10** ¡Error! No se encuentra el origen de la referencia. se muestra los resultados obtenidos, de igual forma en la **Figura 3.11** se presenta la prueba unitaria.



Campo	tipo
Nombre del recurso (campo por defecto)	text
Descripción	text

**Figura 3.10** Página de módulo de Selección de encriptación.

```
//Test de cambio del tipo de encriptacion
it("Test de cambio del tipo de encriptado", async () => {
  const req: ReqCreateResourceType = {
    body: {
      id_user_ref: USER_ID._id,
      name_type_encrypt: "BASE 64"
    },
    headers: {
      usuario_autorizacion: TOKEN
    },
    params: {
      id: RESOURCE_ID
    }
  };
  // Llama al controlador que se encarga de editar el tipo
  await adminChangeTypeEncryptResource(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

**Figura 3.11** Prueba unitaria del módulo de Selección de encriptación.











### 3.2 *Sprint 1* Resultados del módulo de Administrador: Gestión de los tipos de recursos, módulo de registro y modificar perfil.

Las actividades designadas a desarrollarse en este Sprint están conformadas por las siguientes funcionalidades:

- Módulo de administrador: Gestionar los tipos de recurso que puede guardar el usuario.
- Módulo de Registro.
- Módulo de modificar perfil.

**Módulo de administrador: Gestionar los tipos de recurso que puede guardar el usuario.**

El módulo de Administración de los tipos de recursos da la posibilidad de crear nuevas instancias del tipo de recurso que puede guardar el usuario. Este asigna un nombre y se añaden campos los cuales pueden ser de texto, numéricos, de contraseña u ocultos y fechas. Una vez creado este recurso puede ser editado o retirado de la BDD y a su vez este es mostrado al usuario como un tipo de recurso. Con el módulo explicado, en la **Figura 3.12** se muestra el resultado obtenido durante el desarrollo en conjunto con la prueba unitaria que demuestra el correcto funcionamiento y su evidencia en la **Figura 3.13**.

Recurso	Encriptado	Acción
Tipo de contraseña	JWT	 
Tipo de tarjeta	JWT	 
Nota	JWT	 
Identidad	BASE 64	 
Facebook	BASE 64	 

**Figura 3.12** Página del módulo de Administración los recursos.

```
it("Crear nuevo tipo de recurso", async () => {
  bodyCreateResourceAdmin.id_user_ref = USER_ID._id;
  const req: ReqCreateResourceType = {
    body: bodyCreateResourceAdmin,
    headers: {
      usuario_autorizacion: TOKEN
    }
  };
  await adminCreateResource(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura 3.13 Prueba unitaria del módulo de Administración de recursos.

### Módulo de Registro.

El módulo de registro permite al usuario crear una nueva cuenta en el sistema, para lo cual debe completar un formulario en donde se solicita información personal para luego ser almacenada en la BDD. El Administrador al contar con privilegios diferentes este no debe completar un registro como tal, sus credenciales son asignadas por defecto. Completado el registro el sistema se encarga de notificar al correo registrado con un enlace que permite la verificación de la cuenta y el acceso a las funciones habilitadas para el usuario. Como evidencia del desarrollo realizado en este módulo en la **Figura 3.14** se muestra los resultados obtenidos, de igual forma en la **Figura 3.15** se presenta la prueba unitaria.

## CREAR CUENTA

**Nombre**

Nombre

**Apellido**

Apellido

**E-mail**

E-mail

**Fecha de nacimiento**

dd / mm / aaaa



**Contraseña**

\*\*\*\*\*

**Confirmar contraseña**

\*\*\*\*\*

**Registrarse**

[Ya tienes una cuenta?](#)

Figura 3.14 Página del módulo de Registro.

```
// Test de registro de un nuevo usuario
it("Test crear nuevo usuario", async () => {
  const req: any = {
    body: {
      email: MailVerifier.targetUserEmail,
      name: "Test",
      lastname: "User",
      birth: "2000-01-20",
      password: process.env.APP_SERVER_TEST_USER_PASSWORD,
      confirm_password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await registerUser(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura 3.15 Prueba unitaria del módulo de registro.

#### Módulo de modificar perfil.

El sistema brinda *endpoints* para la actualización de los datos ingresados al momento del registro, este módulo está disponible para el usuario siempre que este registrado y tenga una cuenta verificada. La información permitida a actualizar por el usuario y el administrador es: el correo y la contraseña. La información debe ser modificada en la BDD. Los resultados del módulo son representados en la **Figura 3.16** y en la **Figura 3.17** se muestra el resultado de la prueba unitaria.

**E-mail**

**Contraseña**

**Confirmar contraseña**

**Modificar**

Figura 3.16 Página de modificación de perfil – Administrador.

```
//Test de modificacion de perfil de usuario
const req: ReqEditProfileType = {
  body: {
    email: MailVerifier.targetUserEmail,
    password: rollback
      ? process.env.APP_SERVER_TEST_USER_PASSWORD
      : "Nuevapassword123!",
    confirm_password: rollback
      ? process.env.APP_SERVER_TEST_USER_PASSWORD
      : "Nuevapassword123!"
  },
  headers: {
    usuario_autorizacion: TOKEN
  },
  params: {
    id: USER_ID._id
  }
};
await editProfileUser(req, res);
expect(res.status.mock.calls).toEqual([[201]]);
```

Figura 3.17 Prueba unitaria del módulo de modificación del perfil de usuario.

### **3.3 *Sprint 2* Resultados de los módulos de usuario: creación de un recurso, generador de contraseñas y contraseñas robustas.**

Los módulos por desarrollarse en este *Sprint* son:

- Crear una nueva entrada del recurso a autenticar.
- Generar una nueva contraseña segura.
- Modificar los parámetros necesarios para generar una contraseña segura.

#### **Crear una nueva entrada del recurso a autenticar.**

Como usuario registrado y con cuenta verificada, tiene acceso a la creación de recursos con información sensible o que desea mantener privada y protegida. La creación de recursos presenta al usuario una ventana con el tipo de recurso que por defecto son el recurso contraseña, tarjeta y nota, además de los nuevos tipos que el Administrador pueda crear. Cada una de las opciones de tipo de recurso muestra un formulario distinto a completar. Con los campos rellenos la información será almacenada en la BDD para posterior recuperación o actualización según quiera el usuario. Determinada las funcionalidades del módulo, a continuación, se presentan las evidencias de la codificación en la **Figura 3.18** y la prueba unitaria en la **Figura 3.19**.

## Agregar recurso

BASE 64 ▼

**Nombre del recurso**


Ingresar nombre del recurso

**Nombre del campo**

Ingresar nombre del campo

**Tipo del campo**

text ▼



**Cancelar** **Agregar**

Figura 3.18 Página de módulo de creación de nuevo recurso.



```

// Test de creación de un nuevo recurso
it("Test crear nuevo recurso [USER]", async () => {
  const req: ReqCreateResourceType = {
    body: {
      id_resource_admin: RESOURCE_ID,
      id_user_ref: USER_ID._id,
      inputs_data: [
        {
          nombre_del_recurso: "Recurso de prueba User",
          [bodyCreateResourceAdmin.fields[0].label_input]:
            "First test field",
          [bodyCreateResourceAdmin.fields[1].label_input]:
            "Second test field"
        }
      ],
      name_resource_admin: bodyCreateResourceAdmin.name_resource
    },
    headers: {
      usuario_autorizacion: TOKEN
    }
  };
  await userCreateResource(req, res);
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);

```

**Figura 3.19 Prueba unitaria del módulo de creación de un nuevo recurso.**

### Generar una nueva contraseña segura

Dentro de la creación de un nuevo recurso el usuario tiene la funcionalidad de generar una contraseña segura, para lo cual el sistema despliega un modal en donde se presenta la contraseña segura predeterminada, que tiene una longitud de 8 caracteres y formado por caracteres alfanuméricos. La contraseña puede ser copiada para ser probada, caso contrario pasa al formulario de recurso en edición. Con el módulo explicado, en la **Figura 3.20** se muestra el resultado obtenido durante el desarrollo en conjunto con la prueba unitaria que demuestra el correcto funcionamiento y su evidencia en la **Figura 3.21**.

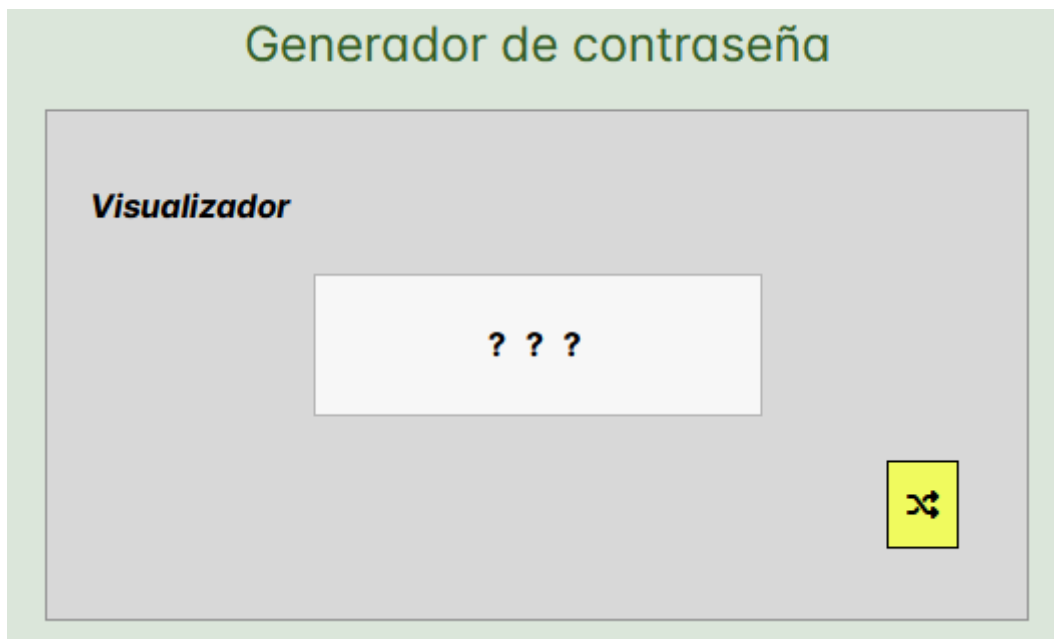


Figura 3.20 Módulo de generación de contraseñas.

```
//Test del generador de contraseñas
test('Genera una nueva contraseña', async () => {
  const wrapper = mount(GeneratePassword)
  wrapper.vm.onGenerateNow()
  // verifica que el password se genero correctamente
  expect(wrapper.vm.password_generate).toBeTruthy()
})
```

Figura 3.21 Prueba unitaria del generador de contraseñas.

### Modificar los parámetros necesarios para generar una contraseña segura

Dentro de la funcionalidad de generar una contraseña segura, no todas las plataformas requieren el mismo formato de contraseña, como es el uso de caracteres especiales. Para dar solución a esta necesidad se muestra la opción de crear contraseñas más robustas o a la medida modificando ciertas opciones como: el uso de mayúsculas, minúsculas, números, caracteres especiales al igual que la cantidad de dichos caracteres aplicando un mínimo o modificando la longitud de la contraseña en general. Determinada las funcionalidades del módulo, a continuación, se presentan las evidencias de la codificación en la **Figura 3.22** y la prueba unitaria en la **Figura 3.23**.

Modo opcional activado

A-Z  a-z  0-9  !@#&\*...

Longitud

0

Generar

Figura 3.22 Módulo de generación de contraseña robusta.

```
// Test para verificar que se generen contraseñas con números
test('Genera una contraseña con numeros y logitus de 10 caracteres', async () => {
  const wrapper = mount(GeneratePassword)
  const obj = {
    target: {
      name: 'checkbox3',
      checked: true
    }
  }
  // cambia la longitud de la prop password
  wrapper.vm.longitud = 10
  wrapper.vm.onOptionalGeneratorPass(obj)
  wrapper.vm.onOptionalGenerateNow(obj)
  const password = wrapper.vm.password_generate
  expect(password.length).toBe(10)
  expect(password).toMatch(/^[0-9]+$/)
})
```

Figura 3.23 Prueba unitaria de las opciones avanzadas del generador de contraseñas.

### 3.4 *Sprint 3* Resultados de los módulos de usuario: modificación de recursos almacenados, visualización de información y selección de tipos de recursos.

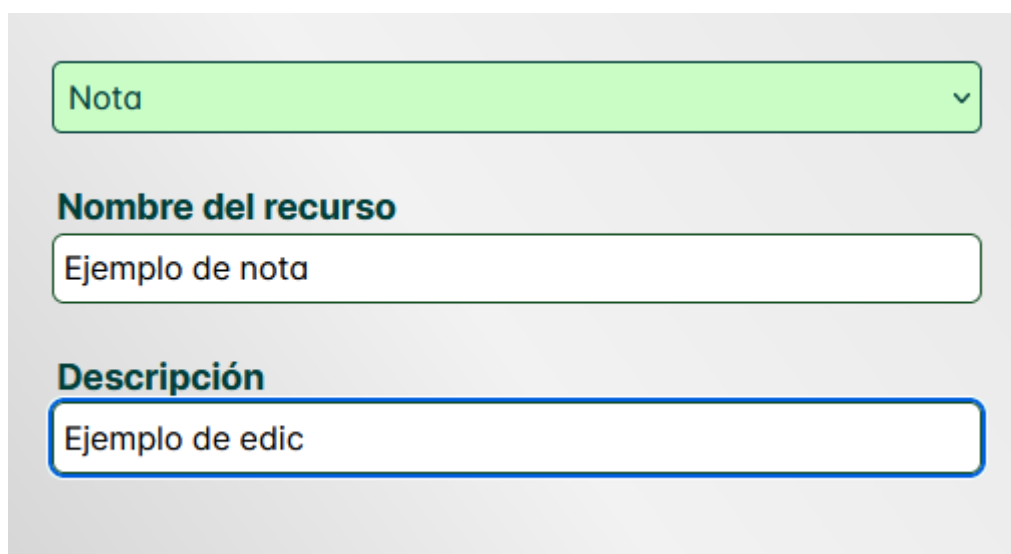
Las actividades para el ultimo sprint de desarrollo son:

- Modificar los recursos almacenados.

- Visualizar la información sensible de los recursos previos.
- Seleccionar el tipo de recursos que desea guardar.

### **Modificar los recursos almacenados**

Para recuperar los recursos almacenados por el usuario el sistema cuenta con *endpoints* dedicados para esta funcionalidad. Los recursos son enlistados en la parte izquierda del tablero del usuario para posterior ser visualizados o modificados. Toda la información de un recurso guardado puede ser modificable a excepción del tipo de recurso. Los recursos que ya no sean necesarios para el usuario pueden ser eliminados del sistema con toda la información almacenada. Como evidencia del desarrollo realizado en este módulo en la **Figura 3.24** se muestra los resultados obtenidos, de igual forma en la **Figura 3.25** se presenta la prueba unitaria.



Nota

**Nombre del recurso**

Ejemplo de nota

**Descripción**

Ejemplo de edic

**Figura 3.24 Edición de recurso guardado.**

```

it('Modificar los recursos almacenados', () => {
  cy.get('a[href="/entries"]', {
    timeout: 5000
  }).click()
  cy.url().should('include', '/entries')
  // Verifica que exista un recurso guardado con el nombre Prueba
  cy.get('div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)').invoke('text').then((text)
=> {
    expect(text).to.equal('Test e2e');
  })
  cy.get('div.dash_content_main > div > div > div table div > button.BTN_WARNING_1').click()
  validateType('input[id="nombre_del_recurso_0"]', 'text')
  cy.get('input[id="nombre_del_recurso_0"]').clear().type('Modificacion de recurso')
  cy.get('div.relative.z-10 div:nth-child(1) > button').click()
  cy.wait(1000)
  cy.get('div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)').invoke('text').then((text)
=> {
    expect(text).to.equal('Modificacion de recurso');
  })
})

```

**Figura 3.25 Test de modificación de recurso.**

### Visualizar la información sensible de los recursos previos

Los recursos recuperados por el usuario una vez que este inicia sesión se muestran en forma de tarjeta en la parte izquierda del panel con la información sensible guardada. Los datos pueden ser copiados por el usuario para ser autenticados o cambiar la visibilidad de cierta información delicada como son las contraseñas o números de tarjetas que siempre se encuentran ocultos al momento de presentarse. Los resultados del módulo son representados en la **Figura 3.26** y en la **Figura 3.27** se muestra el resultado de la prueba unitaria.

## Ver recurso

### Nota

**Nombre del recurso**

**Descripción**

Cancelar

Figura 3.26 Recuperar información de recurso almacenado.

```

it('Visualizar la información sensible de los recursos previos', () => {
  cy.get('a[href="/entries"]', {
    timeout: 5000
  }).click()
  cy.url().should('include', '/entries')
  // Verifica que exista un recurso guardado con el nombre Prueba
  cy.get('div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)').invoke('text').then((text)
=> {
    expect(text).to.equal('Prueba');
  })
  cy.get('div.dash_content_main > div > div > div table div > button.BTN_WARNING_1').click()
  validateType('input[id="nrodetarjeta_2"]', 'password')
  cy.get('div[id="nrodetarjeta"]').click()
  validateType('input[id="nrodetarjeta_2"]', 'text')
  validateType('input[id="codigodeseguridad_5"]', 'password')
  cy.get('div[id="codigodeseguridad"]').click()
  cy.get('div[id="codigodeseguridad"]').click()
  validateType('input[id="codigodeseguridad_5"]', 'text')
})

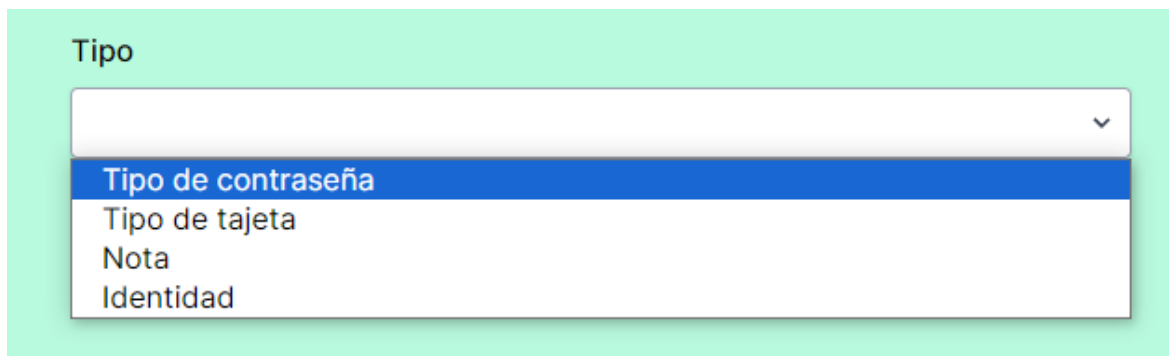
```

Figura 3.27 Test de visualización de los recursos guardados.

### Seleccionar el tipo de recursos que desea guardar

Los tipos de recurso que el usuario puede guardar son una opción modificable en el *combobox* al momento de crear un nuevo recurso, las opciones predeterminadas son el de un recurso tipo contraseña, tarjeta o nota. Los recursos que el

Administrador pueda crear se presentan como una nueva opción elegible por el usuario. De acuerdo con el tipo de recurso seleccionado cambia el formulario y la información que debe proporcionar el usuario. Determinada las funcionalidades del módulo, a continuación, se presentan las evidencias de la codificación en la **Figura 3.28** y la prueba unitaria en la **Figura 3.29**.



**Figura 3.28** Funcionalidad de selección de un tipo de recurso.

```
it('Seleccionar el tipo de recurso que desea guardar', () => {
  cy.get('a[href="/entries"]', {
    timeout: 5000
  }).click()
  cy.url().should('include', '/entries')
  cy.get('div.dash_content_main > div:nth-child(2) > div > button').click()
  // Selecciona el tipo de recurso que desea guardar
  cy.get('div.relative.z-10 > div.fixed.inset-0.z-10.w-screen.overflow-y-auto div > select').select(
    'Tipo de tarjeta')
  // Selecciona el campo de nombre del recurso y escribe Prueba
  cy.get('input[id="nombre_del_recurso_0"]').type('Prueba')
  cy.get('input[id="nombre_tarjeta_1"]').type('Prueba tarjeta')
  cy.get('input[id="nrodetarjeta_2"]').type('112233445566')
  cy.get('input[id="mesdeexpiracion_3"]').type('05')
  cy.get('input[id="aniodeexpiracion_4"]').type('25')
  cy.get('input[id="codigodeseguridad_5"]').type('123')
  cy.get('div.relative.z-10 div:nth-child(1) > button').click()
  cy.get('div.LAYOUT_APP_SIDE_RIGHT > div.dash_content_main > div > div')
    .find('.msg_no_data_stored')
    .should('not.exist');
})
```

**Figura 3.29** Test de selección de tipo de recurso.

### 3.5 Sprint 4 Testeo del sistema *web* y despliegue.

Dado por terminado la codificación del sistema tanto del *frontend* y el *backend* es necesario realizar pruebas antes de desplegar el sistema a producción. Las últimas tareas de este Sprint son:

- Pruebas unitarias y resultados

- Pruebas de carga y estrés
- Pruebas funcionales

### Pruebas unitarias y resultados

Las pruebas unitarias son un tipo de prueba de software en la que se evalúa de manera individual una porción de código pequeña, es importante que este tipo de prueba se realice de manera aislada y controlada para verificar si el resultado obtenido es igual al resultado esperado por el desarrollador. [40]

En la **Figura 3.30** se muestra una porción de código de las pruebas unitarias que corresponde al registro de un nuevo usuario, el resultado se muestra en la **Figura 3.31**. La totalidad de los resultados al igual que el resto de las pruebas se encuentra en el **ANEXO II**.

```
// Test de registro de un nuevo usuario
it("Test crear nuevo usuario", async () => {
  const req: any = {
    body: {
      email: MailVerifier.targetUserEmail,
      name: "Test",
      lastname: "User",
      birth: "2000-01-20",
      password: process.env.APP_SERVER_TEST_USER_PASSWORD,
      confirm_password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await registerUser(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura 3.30 Código de prueba unitaria.



```

PASS test/main.test.ts (84.054 s)
Test Controlador-autenticacion
  Test de registro de nuevo usuario
    ✓ Test crear nuevo usuario (291 ms)
    ✓ Test confirmar cuenta (69631 ms)
  Test Inicio de sesion Administrador
    ✓ Test inicio de sesion [ADMIN] (148 ms)
    ✓ Test cierre de sesion [ADMIN] (12 ms)
  Test Controlador Usuario-Administrador
    ✓ Test de busqueda de usuario [ADMIN] (19 ms)
    ✓ Test de inicio de sesion usuario verificado (113 ms)
    ✓ Test cambio de estado cuenta [ADMIN] inactivo (53 ms)
    ✓ Test de inicio de sesion usuario inactivo (11 ms)
    ✓ Test cambio de estado cuenta [ADMIN] activo (35 ms)
    ✓ Test de inicio de sesion usuario activo (115 ms)
  Test Controlador-Usuario
    Perfil de usuario [USER]
      ✓ Inicio de sesion (116 ms)
      ✓ Editar perfil - Cambiar contraseña (124 ms)
      ✓ Editar perfil - Repetir contraseña (118 ms)
    Test Controlador-Recursos-Usuario
      Tipo de recurso [ADMIN]
        ✓ Inicio de sesion [ADMIN] (120 ms)
        ✓ Validar si el recurso no existe (42 ms)
        ✓ Crear nuevo tipo de recurso (43 ms)
        ✓ Validar si el recurso existe (24 ms)
        ✓ Test de cambio del tipo de encriptado (44 ms)
        ✓ Validar si el recurso cambio de encriptacion (21 ms)
      Recurso de usuario[USER]
        ✓ Inicio de sesion (116 ms)
        ✓ Test crear nuevo recurso [USER] (25 ms)
      Borrar Tipo de recurso [ADMIN]
        ✓ Inicio de sesion (115 ms)
        ✓ Test de eliminar recurso (30 ms)
    Test Eliminar Controlador Usuario-Administrador
      ✓ Test eliminar usuario de prueba [ADMIN] (10 ms)

Test Suites: 1 passed, 1 total
Tests:      24 passed, 24 total
Snapshots: 0 total
Time:      84.395 s

```

**Figura 3.31 Resultado satisfactorio de todas las pruebas unitarias.**

### Pruebas de carga y estrés

Son pruebas de software que miden el comportamiento y reacción de un sistema bajo parámetros de cargar máxima y forzando limites fuera de lo normal, busca como resultado determinar la estabilidad, rendimiento y capacidad del sistema en condiciones difíciles. [41]

Para realizar la prueba de carga y estrés se utiliza la solución de software de Apache *Jmeter*. Al presentar una opción grafica para la configuración de los parámetros necesarios para realizar la prueba es necesario configurar las variables de entorno que se presenta en la **Figura 3.32**.

User Defined Variables		
Name:	Value	Description
RESOURCE_NAME	Nota	El nombre del recurso
RESOURCE_ID	65c0b61a247f26fb1b94d035	El ID del recurso
USER_ID	65c0b37c247f26fb1b94cf81	El ID del usuario
CORREO	michaelle1.1llumiquinga@gmail.com	El correo del usuario
PASSWORD	qawseD5@	La contraseña del usuario

**Figura 3.32 Variables de entorno definidas en Jmeter.**

El resultado de igual forma se presenta de manera grafica en forma de lista o tabla que se muestra en la **Figura 3.33**. ¡Error! No se encuentra el origen de la referencia..

**Figura 3.33 Resultados de la prueba de carga.**

### Pruebas funcionales

Son un tipo de prueba de software que se centra en corroborar que un sistema o componente cumpa con los requerimientos funcionales levantados, la manera de verificar el sistema es por medio de la relación de las entradas y salidas esperadas.

[42]

Para evidenciar las pruebas funcionales en la siguiente **Figura 3.34** se adjunta una captura con los resultados, la totalidad de las pruebas se encuentra en el **ANEXO II**.

Nº	Reportado por: (Persona que realiza prueba)	Aplicación	Rol	Componente	Nivel (alto, medio, bajo)	Incidente	Descripción	Recomendación	23/11/2023	28/11/2023	17/12/2023	29/1/2024	5/2/2024
1	Desarrollador	Sistema Web	Administrador y Usuario	Iniciar y cerrar sesión	alto	Diseño	La pantalla de inicio de sesión no se renderiza	Revisar el routing del inicio de sesión	NO	OK	OK	OK	OK
2	Desarrollador	Sistema Web	Administrador	Cambiar el estado de la cuenta de los usuarios	medio	Funcionalidad	No se actualiza el cambio de estado en la BDD y no se notifica el cambio por correo.	Modificar el servicio de correo de Google cloud y actualizar el token	NO	OK	OK	OK	OK
3	Desarrollador	Sistema Web	Administrador	Seleccionar el tipo de encriptación de los datos.	alto	Funcionalidad	No hay cambio de encriptación	Emplear otra librería de encriptación	NO	NO	OK	OK	OK
4	Desarrollador	Sistema Web	Administrador	Gestionar los tipos de recursos que puede guardar el usuario	medio	Funcionalidad	No se reinicia la creación de un nuevo recurso	Reiniciar el arreglo del la creación de un nuevo recurso	NO	NO	OK	OK	OK
5	Desarrollador	Sistema Web	Usuario	Registrarse	alto	Funcionalidad	No se envía el correo de verificación	Revisar el servicio de Google cloud. Actualizar el token de google	NO	NO	OK	OK	OK
6	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso a agregar	alto	Funcionalidad	No se crea el nuevo recurso	Revisar la funcionalidad del envío de datos al backend	NO	OK	OK	OK	OK
7	Desarrollador	Sistema Web	Usuario	Modificar el perfil de usuario	alto	Funcionalidad	No se actualiza la información del correo y la contraseña	Revisar la funcionalidad de actualización del correo	NO	NO	OK	OK	OK
8	Desarrollador	Sistema Web	Usuario	Generar una nueva contraseña	alto	Funcionalidad	Contraseña predeterminada muy pequeña	Revisar el componente del generador de contraseñas en el frontend	NO	OK	OK	OK	OK

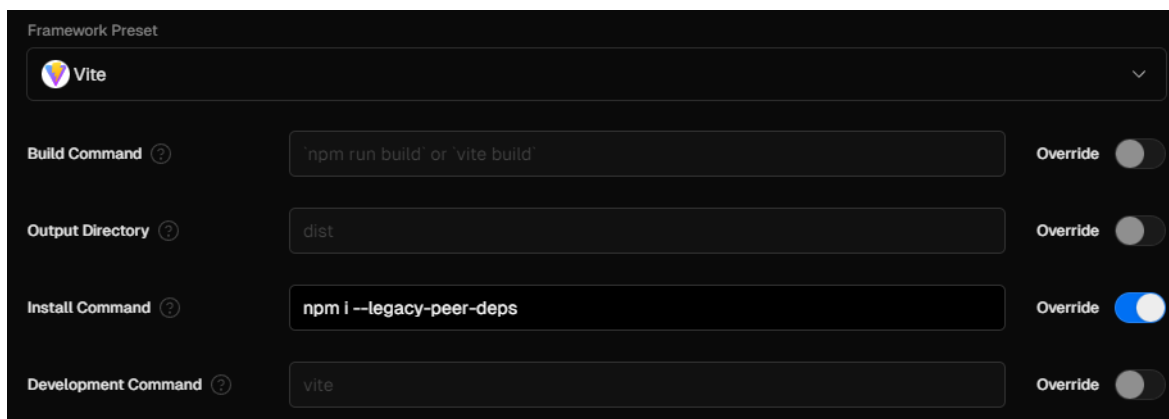
**Figura 3.34 Pruebas funcionales del sistema.**

### Despliegue del sistema *web*

Terminada la codificación de todo el sistema y la comprobación de los módulos con el testeo, la parte final del proyecto es llevar a producción en un *hosting* tanto del *backend* como el *frontend*.

Para realizar el despliegue del sistema se hace el uso de dos plataformas con planes gratuitos: Vercel y Koyeb.

Vercel es una plataforma dedicada en su mayor parte al despliegue de proyectos desarrollados en Javascript para el *frontend*. Por su parte el *backend* es desplegado en Koyeb, plataforma centrada en el despliegue de proyectos de lado del servidor. En la **Figura 3.35** se muestra parte de la configuración realizada en Vercel y en la **Figura 3.36** se muestra de igual forma la configuración de Koyeb.



**Figura 3.35 Modificación de comando de instalación en Vercel.**

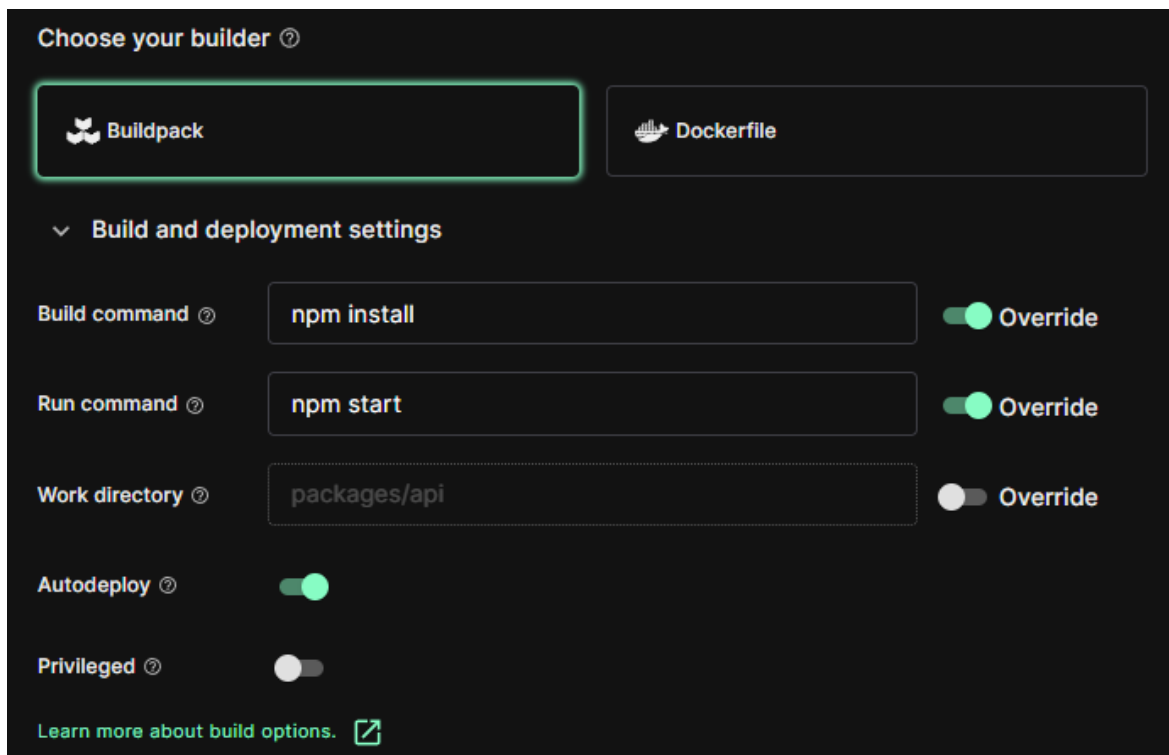


Figura 3.36 Modificación de los comandos de instalación e inicio en Koyeb.

## 4 CONCLUSIONES

Como penúltimo punto se enlistan las conclusiones obtenidas en el transcurso del desarrollo del sistema.

- La metodología Scrum facilitó al cumplimiento con los requerimientos del sistema, por lo que ayuda de manera considerable a cumplir con los tiempos y los objetivos propuestos al inicio del proyecto.
- El modelado de las pantallas, modales y diferentes elementos visuales por medio de herramientas de diseño Figma, ayudó a obtener una visión preliminar del funcionamiento del sistema, logrando tener una visión clara del objetivo del proyecto.
- Las herramientas de software empleadas en la etapa de desarrollo del código juegan un papel importante en la codificación puesto que ayuda en la codificación, la interacción con el sistema en fases iniciales, el contar con opciones dedicadas al testeo y demás ayudas al desarrollador.
- El desarrollo del *backend* en conjunto con el OMD como es Mongoose facilitó la comunicación con la BDD y todas las interacciones necesarias para los diferentes módulos y funcionalidades propuestas.
- El *frameworks* de CSS de TailwindCSS facilita el estilado del proyecto, de tal manera que el desarrollo del *frontend* mejora de manera considerable en cuestión del tiempo.
- Se comprobó el buen funcionamiento de cada uno de los módulos, gracias a la realización de las pruebas funcionales; además, se verificó que los perfiles se establecieron correctamente, obteniendo un software de calidad.
- El despliegue del sistema fue rápido e intuitivo en las plataformas de Vercel y Koyeb gracias a que se vincularon con el repositorio y se modificaron las variables de entorno de los proyectos.

## **5 RECOMENDACIONES**

Como último punto se enlistan las recomendaciones presentadas en el tiempo de desarrollo del sistema.

- Se recomienda la adición de nuevas funcionalidades de seguridad importantes hacia el usuario como son el factor de doble autenticación al igual que el uso de información biométrica para el uso del sistema.
- En dado caso de que el sistema lo requiera y se necesite añadir más de una cuenta de Administrador es necesario acceder de manera manual a la base de datos y cambiar la información correspondiente.
- Teniendo en cuenta el objetivo del sistema se recomienda realizar respaldos de la BDD de manera recurrente.

## 6 REFERENCIAS BIBLIOGRÁFICAS

- [1] «Incibe,» 06 02 2019. [En línea]. Available: <https://www.incibe.es/ciudadania/blog/sabias-que-el-90-de-las-contrasenas-son-vulnerables#:~:text=Muchos%20usuarios%20que%20navegan%20por,peligros%20que%20esta%20pr%C3%A1ctica%20conlleva..> [Último acceso: 22 11 2023].
- [2] J. Á. Gómez, «Semantic systems,» [En línea]. Available: <https://www.semantic-systems.com/semantic-noticias/articulos-tecnologicos/los-peligros-de-las-contrasenas-y-sus-soluciones/>. [Último acceso: 22 11 2023].
- [3] F. Gonzáles, «Wired,» 04 05 2023. [En línea]. Available: <https://es.wired.com/articulos/las-contrasenas-mas-usadas-en-mexico-y-otros-paises-de-latam#:~:text=mayo%20de%202023-,Las%20contrase%C3%B1as%20m%C3%A1s%20usadas%20en%20M%C3%A9xico%20y%20otros%20pa%C3%ADses%20de,un%20pendiente%20en%20toda%20LATAM..> [Último acceso: 22 11 2023].
- [4] J. García, «Xataka,» 15 01 2023. [En línea]. Available: <https://www.xataka.com/seguridad/estamos-2023-123456-sigue-siendo-contrasena-usada-hablemos-como-crear-gestionar-contrasena-segura>. [Último acceso: 22 11 2023].
- [5] D. Borovskoy, «LinkedIn,» 27 09 2023. [En línea]. Available: <https://es.linkedin.com/pulse/que-es-backend-caracter%C3%ADsticas-2023-denis-borovskoy>. [Último acceso: 15 11 2023].
- [6] «Red Hat,» Red Hat, 31 07 2023. [En línea]. Available: <https://www.redhat.com/es/topics/api/what-is-a-rest-api>. [Último acceso: 15 11 2023].
- [7] «IMB,» IMB, [En línea]. Available: <https://www.ibm.com/mx-es/topics/relational-databases>. [Último acceso: 15 11 2023].
- [8] I. J. B. García, «Servnet,» 30 03 2021. [En línea]. Available: <https://www.servnet.mx/blog/backend-y-frontend-partes-fundamentales-de-la-programacion-de-una-aplicacion-web>. [Último acceso: 16 05 2023].
- [9] «NeoAttack,» 27 08 2020. [En línea]. Available: <https://neoattack.com/neowiki/framework/>. [Último acceso: 16 05 2023].
- [10] M. N. J. R. R. Fielding, 06 2022. [En línea]. Available: <https://www.rfc-editor.org/rfc/rfc9110.html#name-references>. [Último acceso: 23 05 2023].
- [11] A. Núñez, «Businessinsider,» 11 05 2021. [En línea]. Available: <https://www.businessinsider.es/api-sirve-todo-necesitas-saber-861403>. [Último acceso: 23 05 2023].
- [12] R. A. Tab Atkins Jr., «W3,» 18 12 2020. [En línea]. Available: <https://www.w3.org/TR/css-grid-1/>. [Último acceso: 22 05 2023].

- [13] J. Nielsen, «Nielsen Norman Group,» 15 11 2020. [En línea]. Available: <https://www.nngroup.com/articles/ten-usability-heuristics/>. [Último acceso: 22 05 2023].
- [14] J. N. Don Norman, «Nielsen Norman Group,» [En línea]. Available: <https://www.nngroup.com/articles/definition-user-experience/>. [Último acceso: 22 05 2023].
- [15] S. Gatica, «Easytax,» [En línea]. Available: <https://www.easytax.cl/blog/modelos-de-encryptacion-que-son-y-cuantos-tipos-hay/>. [Último acceso: 03 12 2023].
- [16] «Ionos,» 22 02 2023. [En línea]. Available: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/hash/#:~:text=El%20hashing%20es%20un%20m%C3%A9todo,valores%20originales%20sin%20una%20clave.> [Último acceso: 03 12 2023].
- [17] «Microsoft learn,» [En línea]. Available: <https://learn.microsoft.com/es-es/visualstudio/test/unit-test-basics?view=vs-2022>. [Último acceso: 22 05 2023].
- [18] T. Hamilton, «Guru99,» 13 05 2023. [En línea]. Available: <https://www.guru99.com/load-testing-tutorial.html>. [Último acceso: 22 05 2023].
- [19] A. Rios, «Encora,» 13 01 2021. [En línea]. Available: <https://www.encora.com/es/blog/pruebas-de-rendimiento-cuando-y-como>. [Último acceso: 22 05 2023].
- [20] R. K. Yin, Case Study Research: Design and Methods, London: SAGE, 2009.
- [21] R. C. Martin, Agile Software Development, Principles, Patterns, and Practices, Pearson, 2002.
- [22] L. Gonçalves, «ADAPT methodology,» 28 11 2021. [En línea]. Available: <https://adaptmethodology.com/es/que-es-la-metodologia-agil/>. [Último acceso: 22 05 2023].
- [23] «Agiles Scrum,» 06 12 2022. [En línea]. Available: <https://www.agilescrum.cl/post/que-es-scrum>. [Último acceso: 23 05 2023].
- [24] C. MacNeil, «Asana,» 03 01 2023. [En línea]. Available: <https://asana.com/es/resources/product-owner>. [Último acceso: 23 05 2023].
- [25] M. Á. d. Dios, «Wam Global Growth Agents,» 09 05 2022. [En línea]. Available: <https://www.waremarketing.com/es/blog/metodologia-scrum-que-es-y-como-funciona.html>. [Último acceso: 23 05 2023].
- [26] V. Singh, «Tools A,» 07 07 2021. [En línea]. Available: <https://www.toolsqa.com/agile/scrum/scrum-development-team/#:~:text=on%20Development%20Team,Who%20is%20the%20SCRUM%20Development%20Team%3F,and%20meeting%20the%20sprint%20goal.> [Último acceso: 23 05 2023].



- [27] C. Harris, «Atlassian,» [En línea]. Available: <https://www.atlassian.com/es/agile/scrum/artifacts#:~:text=del%20scrum%20%C3%A1gil%3F-,Los%20artefactos%20del%20scrum%20%C3%A1gil%20son%20informaci%C3%B3n%20que%20un%20equipo,tareas%20realizadas%20durante%20el%20proyecto..> [Último acceso: 23 05 2023].
- [28] «Asana,» 13 11 2022. [En línea]. Available: <https://asana.com/es/resources/requirements-gathering>. [Último acceso: 23 05 2023].
- [29] «Mountain Goat Software,» [En línea]. Available: <https://www.mountaingoatsoftware.com/agile/user-stories>. [Último acceso: 23 05 2023].
- [30] «Scrum.org,» [En línea]. Available: <https://www.scrum.org/resources/what-is-a-product-backlog>. [Último acceso: 23 04 2023].
- [31] A. Pérez, «OBS Business School,» 25 04 2021. [En línea]. Available: <https://www.obsbusiness.school/blog/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>. [Último acceso: 23 05 2023].
- [32] C. Busquets, «Uifrommars,» [En línea]. Available: <https://www.uifrommars.com/figma-primeros-pasos/#conclusiones-figma>. [Último acceso: 05 12 2023].
- [33] M. Fowler, «Martin Fowler,» 1 08 2019. [En línea]. Available: <https://martinfowler.com/architecture/>. [Último acceso: 23 05 2023].
- [34] I. Garcia, «Caront estudio,» 30 11 2021. [En línea]. Available: <https://carontestudio.com/blog/que-es-modelo-vista-controlador/>. [Último acceso: 23 05 2023].
- [35] «Indeed,» 07 02 2023. [En línea]. Available: <https://mx.indeed.com/orientacion-profesional/desarrollo-profesional/herramientas-desarrollo-software>. [Último acceso: 07 06 2023].
- [36] A. Barragán, «OpenWebinars,» 26 11 2021. [En línea]. Available: <https://openwebinars.net/blog/que-es-vue-js-y-que-lo-diferencia-de-otros-frameworks/>. [Último acceso: 23 05 2023].
- [37] C. Moreno, «Medium,» 07 06 2018. [En línea]. Available: <https://medium.com/@khriztianmoreno/vue-cli-la-forma-m%C3%A1s-r%C3%A1pida-de-comenzar-con-vue-js-2aed33d3ee2e>. [Último acceso: 23 05 2023].
- [38] Parzibyte, «Parzibyte,» 23 05 2019. [En línea]. Available: <https://parzibyte.me/blog/2019/05/23/primeros-pasos-vue-router-enrutador-vue/#:~:text=Vue%20Router%20es%20el%20enrutador,de%20banda%20y%20agregando%20velocidad..> [Último acceso: 23 05 2023].
- [39] «QAlified,» 09 08 2023. [En línea]. Available: <https://qalified.com/es/blog/postman-para-api-testing/>. [Último acceso: 15 11 2023].

- [40] «Amazon,» Amazon, [En línea]. Available: <https://aws.amazon.com/es/what-is/unit-testing/#:~:text=Una%20prueba%20unitaria%20es%20un,la%20l%C3%B3gica%20te%C3%B3rica%20del%20desarrollador..> [Último acceso: 28 01 2024].
- [41] «Atentus,» 10 11 2022. [En línea]. Available: <https://atentus.com/2022/11/10/pruebas-de-carga/>. [Último acceso: 28 01 2024].
- [42] A. Z. Chernyak, «Zaptest,» [En línea]. Available: <https://www.zaptest.com/es/que-son-las-pruebas-funcionales-tipos-ejemplos-lista-de-comprobacion-y-aplicacion#:~:text=Las%20pruebas%20funcionales%20son%20una,despu%C3%A9s%20de%20iniciar%20la%20sesi%C3%B3n..> [Último acceso: 28 01 2024].
- [43] L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.
- [44] E. Equipo editorial, «Concepto.de,» 05 08 2021. [En línea]. Available: <https://concepto.de/html/>. [Último acceso: 18 05 2023].
- [45] G. B., «Hostinger,» 11 01 2023. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-css>. [Último acceso: 17 05 2023].
- [46] «Kinsta,» 10 02 2023. [En línea]. Available: <https://kinsta.com/es/base-de-conocimiento/que-es-javascript/>. [Último acceso: 17 05 2023].
- [47] J. Román, «lenguajejs.com,» [En línea]. Available: <https://lenguajejs.com/vuejs/introduccion/que-es-vue/>. [Último acceso: 18 05 2023].
- [48] C. Murphy, «Smashing magazine,» 21 02 2018. [En línea]. Available: <https://www.smashingmagazine.com/2018/02/comprehensive-guide-ui-design/>. [Último acceso: 23 05 2023].
- [49] Jesús, «Dongee,» 16 03 2023. [En línea]. Available: <https://www.dongee.com/tutoriales/figma-que-es/>. [Último acceso: 23 05 2023].
- [50] A. Camus, «Microverse,» 05 05 2022. [En línea]. Available: <https://www.microverse.org/blog/javascript-library-vs-javascript-frameworks-the-differences>. [Último acceso: 23 02 2023].
- [51] M. A. Alvarez, «Desarrolloweb,» 14 09 2018. [En línea]. Available: <https://desarrolloweb.com/articulos/axios-ajax-cliente-http-javascript.html>. [Último acceso: 23 05 2023].
- [52] A. Barragán, «OpenWebinars,» 10 12 2021. [En línea]. Available: <https://openwebinars.net/blog/vuejs-vuetify-crea-interfaces-de-usuario-reutilizables/>. [Último acceso: 23 05 2023].
- [53] M. Iriarte, «Somos PNT,» 09 01 2023. [En línea]. Available: <https://sospnt.com/blog/302-estado-global-con-vuex-en-vue-js#:~:text=Api%20en%20Vue->

,%C2%BFQu%C3%A9%20es%20VueX%3F,puede%20modificar%20de%20forma%20predecible.. [Último acceso: 23 05 2023].

## **7 ANEXOS**

En esta sección se muestran los cuatro anexos del proceso previo al desarrollo del componente sistema.

- Anexo I. Porcentaje del Turnitin
- Anexo II. Manual de técnico
- Anexo III. Manual de usuario
- Anexo IV. Manual de instalación

## ANEXO I

F\_AA\_236

### CERTIFICADO DE ORIGINALIDAD TRABAJO DE INTEGRACIÓN CURRICULAR

Quito, D.M. 21 de febrero de 2024

De mi consideración:

Yo, LORENA ELIZABETH CHULDE OBANDO, en calidad de Director del Trabajo de Integración Curricular titulado FRONTEND asociado al "DESARROLLO DEL SISTEMA WEB DE GESTIÓN DE CONTRASEÑAS." elaborado por el estudiante VALENZUELA SANDOVAL MICHAELLE AURELIO de la carrera en TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta antiplagio "TURNITIN", he solicitado a la Biblioteca General el informe para la revisión de originalidad del documento escrito producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 10%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

[informe-turnitin - OneDrive \(sharepoint.com\)](#)

Atentamente,



**Lorena Elizabeth Chulde Obando**  
**Docente**  
**Escuela de Formación de Tecnólogos**

## ANEXO II

### Recopilación de requerimientos

En la **Tabla I** se detallan todos los requerimientos tomados en cuenta de principio a fin para el desarrollo de este proyecto.

**Tabla I Recopilación de requerimientos completa.**

RECOPIACIÓN DE REQUERIMIENTOS		
TIPO DEL SISTEMA	ID - RR	ENUNCIADO DEL ÍTEM
	<b>RR001</b>	El administrador y el usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Iniciar y cerrar sesión</li> </ul>
	<b>RR002</b>	El administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Cambiar el estado de la cuenta de los usuarios (activo a inactivo)</li> </ul>
	<b>RR003</b>	El administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Seleccionar el tipo de encriptación de los datos.</li> </ul>
	<b>RR004</b>	El administrador necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Gestionar los tipos de recurso que puede guardar el usuario.</li> </ul>
	<b>RR005</b>	El usuario necesita consumir varios <i>endpoints</i> para <ul style="list-style-type: none"> <li>• Registrarse</li> </ul>

	<b>RR006</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Crear una nueva entrada del recurso a autenticar.</li> </ul>
	<b>RR007</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Modificar el perfil de usuario.</li> </ul>
	<b>RR008</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Generar una nueva contraseña segura.</li> </ul>
	<b>RR009</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Modificar los parámetros necesarios para generar una contraseña segura.</li> </ul>
	<b>RR010</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Modificar los recursos almacenados.</li> </ul>
	<b>RR011</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Visualizar la información sensible de los recursos previos.</li> </ul>
	<b>RR012</b>	El usuario necesita consumir varios <i>endpoints</i> para: <ul style="list-style-type: none"> <li>• Seleccionar el tipo de recurso que desea guardar.</li> </ul>

## Historias de usuario

En este apartado se detallan cada una de las historias de usuario las cuales fueron elaboradas en base a las necesidades del cliente.

**Tabla II Historia de usuario 001.**

HISTORIA DE USUARIO	
<b>ID:</b> HU-001	<b>Usuario:</b> Administrador y usuario
<b>Nombre de la Historia:</b> Iniciar y cerrar sesión	
<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Medio
<b>Desarrollador responsable:</b> Michaelle Valenzuela	

<p><b>Descripción:</b></p> <p><b>Como:</b> Administrador y usuario</p> <p><b>Quiero:</b> Iniciar y cerrar sesión en el sistema</p> <p><b>Para:</b> Permitir el acceso a recursos y privilegios según el rol asignado</p>
<p><b>Alcance:</b></p> <ol style="list-style-type: none"> <li>1. El botón “Registrarse” dentro de la ruta de “Registro” abrirá una pestaña que tendrá un formulario con los siguientes campos: <ol style="list-style-type: none"> <li>a. Nombre</li> <li>b. Correo Electrónico</li> <li>c. Contraseña</li> <li>d. Repetir contraseña</li> </ol> </li> <li>2. Botón “Crear cuenta” que permitirá guardar la información del usuario a ser creado.</li> </ol>
<p><b>Observación:</b></p> <ul style="list-style-type: none"> <li>• <b>Nombre</b> (campo de texto): nombre del usuario a guardar.</li> <li>• <b>Correo electrónico</b> (campo de texto): debe cumplir con el formato de correo válido.</li> <li>• <b>Contraseña</b> (campo de contraseña): contraseña de un mínimo de 8 caracteres.</li> <li>• <b>Repetir contraseña</b> (campo de contraseña): repetir la contraseña ingresada en el campo anterior.</li> <li>• <b>Botón “Crear cuenta”:</b> acción que enviará de forma automática un correo de notificación al usuario de su creación de cuenta. El correo deberá contar con el link de verificación de la cuenta que envía a una ruta con el mensaje de Verificar cuenta y redirige a la ventana de <i>Login</i>. La contraseña deberá tener un mínimo de 8 caracteres entre caracteres alfanuméricos y caracteres especiales. En caso de olvido de la contraseña se cuenta con la opción de “Olvido su contraseña” el cual permite crear una contraseña nueva siempre y cuando se verifique que el correo al que desea recuperar es el del usuario.</li> </ul>
<p><b>Criterios de Aceptación:</b></p> <ul style="list-style-type: none"> <li>• Todos los campos (Nombre, Correo, Contraseña, Repetir contraseña) son obligatorios.</li> <li>• No se permitirá la creación de un usuario si no se tiene todos los campos completos.</li> <li>• Se presentará mensajes de advertencia por cada uno de los campos, si los mismos no cumplen con lo establecido.</li> <li>• Se presentará un mensaje a modo de alerta cuando el usuario se ha creado de forma satisfactoria.</li> </ul>

**Tabla III Historia de usuario 002.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-002	<b>Usuario:</b> Administrador



<b>Nombre de la Historia:</b> Cambiar el estado de la cuenta de los usuarios	
<b>Prioridad para el negocio:</b> Medio	<b>Riesgo para desarrollo:</b> Medio
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b> <b>Como:</b> Administrador <b>Quiero:</b> Modificar el estado de una cuenta de usuario <b>Para:</b> Permitir o negar el acceso al sistema	
<b>Alcance:</b> <ol style="list-style-type: none"> <li>1. Se recupera todos los usuarios registrados en la BDD.</li> <li>2. Se presenta el nombre del usuario junto con un switch el cual modifica el estado de la cuenta.</li> <li>3. El switch cambia el estado modificando la información en la BDD.</li> </ol>	
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Switch:</b> al ser activado modificara de manera inmediata en la BDD el valor de estado el cual por defecto se encuentra en "true" y cambiarlo a "false" negando el acceso del usuario al sistema.</li> </ul>	
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>• La recuperación de los usuarios será de todos aquellos que tengan una cuenta registrada de manera correcta.</li> <li>• El cambio de estado se realizará de manera inmediata en la BDD.</li> </ul>	

**Tabla IV Historia de usuario 003.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-003	<b>Usuario:</b> Administrador
<b>Nombre de la Historia:</b> Seleccionar el tipo de encriptación de los datos	
<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Alto
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b> <b>Como:</b> Administrador <b>Quiero:</b> Modificar el tipo de encriptación que se usa en el sistema <b>Para:</b> Cambiar el tipo de encriptación de los recursos que se almacenan por el usuario	
<b>Alcance:</b> <ol style="list-style-type: none"> <li>1. El combobox presenta tres opciones de encriptación de la información.</li> <li>2. Al momento de seleccionar la forma de encriptación de los datos, el cambio se realiza de inmediato en la BDD.</li> </ol>	
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Seleccionar</b> (<i>comboBox</i>): tipo de encriptación a utilizar en el almacenamiento del recurso del sistema.</li> </ul>	

**Criterios de Aceptación:**

- El *comboBox* cambia de manera inmediata el tipo de encriptación en la BDD.

**Tabla V Historia de usuario 004.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-004	<b>Usuario:</b> Administrador
<b>Nombre de la Historia:</b> Gestionar los tipos de recurso que puede guardar el usuario	
<b>Prioridad para el negocio:</b> Medio	<b>Riesgo para desarrollo:</b> Medio
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b>	
<b>Como:</b> Administrador	
<b>Quiero:</b> Crear un nuevo tipo de recurso	
<b>Para:</b> Ampliar las opciones que tiene el usuario con respecto a los recursos que puede guardar	
<b>Alcance:</b>	
<ol style="list-style-type: none"> <li>1. El botón "Crear nuevo recurso" dentro del panel del Administrador abrirá un modal en el cual se puede parametrizar las entradas para crear un nuevo recurso.</li> <li>2. Botón "Guardar" que permitirá guardar los parámetros del nuevo recurso a crear.</li> <li>3. Botón "Cancelar" cierra el modal y descarta toda la información ingresada.</li> </ol>	
<b>Observación:</b>	
<ul style="list-style-type: none"> <li>• <b>Botón "Crear nuevo recurso":</b> abrirá un modal en donde se puede parametrizar los datos de ingreso para crear un nuevo recurso.</li> <li>• <b>Botón "Guardar":</b> enviara la nueva información a la BDD en donde se guarda y se muestra como una posibilidad de almacenamiento de recurso.</li> <li>• <b>Botón "Cancelar":</b> cierra el modal y descarta la información ingresada para registrar un nuevo tipo de recurso.</li> </ul>	
<b>Criterios de Aceptación:</b>	
<ul style="list-style-type: none"> <li>• Todos los campos que sean agregados por el administrador serán guardados en la base de datos.</li> <li>• Una vez guardado el nuevo tipo de recurso, este se encontrará disponible para ser usado por el usuario de manera inmediata.</li> </ul>	

**Tabla VI Historia de usuario 005.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-005	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Registrarse	

<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Bajo
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b> <b>Como:</b> Usuario <b>Quiero:</b> Crear una nueva cuenta de usuario <b>Para:</b> Tener acceso al sistema, funciones y recursos disponibles.	
<b>Alcance:</b> <ol style="list-style-type: none"> <li>1. En la ruta de registro se presenta el formulario con la siguiente información a completar: <ol style="list-style-type: none"> <li>a. Nombre.</li> <li>b. Correo electrónico.</li> <li>c. Contraseña.</li> <li>d. Repetir contraseña.</li> </ol> </li> <li>2. Botón "Crear cuenta" que permitirá guardar la información del usuario en la BDD.</li> </ol>	
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Nombre</b> (campo de texto): Nombre del usuario.</li> <li>• <b>Correo electrónico</b> (campo de texto): debe cumplir con el formato de correo válido.</li> <li>• <b>Contraseña</b> (campo de texto): contraseña con un mínimo de 8 caracteres.</li> <li>• <b>Repetir contraseña</b> (campo texto): repetir la contraseña ingresada en el campo de "Contraseña", esta debe ser la misma para ser valida.</li> <li>• <b>Botón "Crear cuenta"</b>: acción que enviará de forma automática un correo de notificación al usuario para validar su cuenta. El correo deberá contar con el link que dirige a la ventana de verificación de cuenta, mismo que pasado 5 segundo redirige a la página de <i>Login</i>.</li> </ul>	
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>• Todos los campos (Nombre, Correo electrónico, Contraseña y Repetir contraseña) son obligatorios.</li> <li>• No se permitirá la creación de un usuario si no se tiene todos los campos completos.</li> <li>• Se presentará mensajes de advertencia por cada uno de los campos, si los mismos no cumplen con lo establecido.</li> <li>• Se presentará un mensaje a modo de alerta cuando el usuario se ha creado de forma satisfactoria.</li> </ul>	

**Tabla VII Historia de usuario 006.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-006	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Crear una nueva entrada del recurso a autenticar	
<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Bajo

<b>Desarrollador responsable:</b> Michaelle Valenzuela
<b>Descripción:</b> <b>Como:</b> Usuario <b>Quiero:</b> Crear un nuevo recurso <b>Para:</b> Guardar información sensible que a posterior servirá para autenticación
<b>Alcance:</b> <ol style="list-style-type: none"> <li>1. En la ruta de recursos, el botón “Nuevo recurso” mostrar un formulario con los siguientes campos a completar: <ol style="list-style-type: none"> <li>a. Tipo</li> <li>b. Nombre</li> <li>c. Usuario</li> <li>d. Contraseña</li> </ol> </li> <li>2. Botón “Guardar” que permitirá guardar la información del nuevo recurso.</li> <li>3. Botón “Cancelar” que permitirá descartar la información ingresada por el usuario.</li> </ol>
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Tipo</b> (combobox): permitirá seleccionar el tipo de recurso a guardar.</li> <li>• <b>Nombre</b> (campo de texto): nombre del recurso que se almacenara.</li> <li>• <b>Usuario</b> (campo de texto): correo electrónico o usuario del sistema o plataforma a autenticar.</li> <li>• <b>Contraseña</b> (campo de contraseña): contraseña de un mínimo de 8 caracteres a autenticar.</li> <li>• <b>Botón “Nuevo recurso”</b>: acción que desplegara un formulario con los campos de tipo, nombre, usuario y contraseña.</li> <li>• <b>Botón “Guardar”</b>: acción que guardar la información en la BDD que a posterior podrá ser editado o recuperado por el usuario.</li> <li>• <b>Botón “Cancelar”</b>: acción que descartara toda la información ingresa por el usuario y oculta el formulario de crear un nuevo recurso.</li> </ul>
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>• El campo (Nombre) es obligatorio.</li> <li>• No se permitirá la creación de un recurso si no se tiene el campo completo.</li> <li>• Se presentará un mensaje de error cuando el recurso no ha sido creado, junto con el porque de la no ejecución de dicha acción.</li> <li>• Se presentará un mensaje a modo de alerta cuando el recurso se ha guardado de forma satisfactoria.</li> </ul>

**Tabla VIII Historia de usuario 007.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-007	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Modificar el perfil de usuario	
<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Bajo

<b>Desarrollador responsable:</b> Michaelle Valenzuela
<b>Descripción:</b> <b>Como:</b> Usuario <b>Quiero:</b> Modificar el perfil de usuario <b>Para:</b> Para actualizar la información ingresada durante el registro
<b>Alcance:</b> <ol style="list-style-type: none"> <li>4. El botón “Modificar perfil” dentro de los recursos del usuario abrirá una ruta con un formulario con los siguientes campos: <ol style="list-style-type: none"> <li>d. Correo</li> <li>e. Contraseña</li> <li>f. Repita la nueva contraseña</li> </ol> </li> <li>5. Botón “Guardar” que permitirá guardar la información del usuario a ser modificada.</li> <li>6. Botón “Cancelar” que permitirá deshacer (cerrar la página) de modificar el perfil de usuario.</li> </ol>
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Correo</b> (campo de texto): debe cumplir con el formato de correo valido.</li> <li>• <b>Contraseña</b> (campo de texto): debe tener un mínimo de 8 caracteres y no debe ser igual a la contraseña ya guardada.</li> <li>• <b>Repita la nueva contraseña</b> (campo de texto): repetir la contraseña ingresada en el campo de “Contraseña”, esta debe ser la misma para ser valida.</li> <li>• <b>Botón “Guardar”</b>: que actualizara de manera inmediata la nueva información en la BDD.</li> <li>• <b>Botón “Cancelar”</b>: que descartara la información ingresada por el usuario y cerra la página de modificar el perfil de usuario.</li> </ul>
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>• Se presentará un mensaje de error cuando la actualización no se realizó de manera satisfactoria, junto con el porque de la no ejecución de dicha acción.</li> <li>• Se presentará un mensaje a modo de alerta cuando la información se actualizo de forma satisfactoria.</li> </ul>

**Tabla IX Historia de usuario 008.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-008	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Generar una nueva contraseña segura	
<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Medio
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b>	
<b>Como:</b> Usuario	

<b>Quiero:</b> Generar una nueva contraseña
<b>Para:</b> Completar el formulario del recurso con una contraseña segura
<b>Alcance:</b> <ol style="list-style-type: none"> <li>1. El botón “Generar contraseña” dentro de los recursos del usuario despliega un modal con la siguiente información: <ol style="list-style-type: none"> <li>a. Un campo de texto donde se generará una contraseña segura con los parámetros predeterminados (contraseña alfanumérica con una longitud de 8 caracteres)</li> </ol> </li> <li>2. Botón “Seleccionar” que permitirá colocar la nueva contraseña segura en el formulario a completar.</li> <li>3. Botón “Cancelar” que permitirá deshacer (cerrar el modal) y descartar la contraseña segura.</li> </ol>
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Contraseña</b> (campo de texto): se genera una contraseña segura con los parámetros predeterminados, contraseña alfanumérica con una longitud de 8 caracteres.</li> <li>• <b>Botón “Seleccionar”</b>: acción que pasara la contraseña mostrada al formulario del nuevo recurso a guardarse.</li> <li>• <b>Botón “Cancelar”</b>: acción que descarta la contraseña presentada y cierra el modal para continuar con los demás campos del formulario.</li> </ul>
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>• Se presentará un mensaje a modo de alerta cuando la contraseña sea seleccionada.</li> </ul>

**Tabla X Historia de usuario 009.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-009	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Modificar los parámetros necesarios para generar una contraseña	
<b>Prioridad para el negocio:</b> Medio	<b>Riesgo para desarrollo:</b> Alto
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b>	
<b>Como:</b> Usuario	
<b>Quiero:</b> Modificar los parámetros del generador de contraseñas	
<b>Para:</b> Crear una contraseña con mayor seguridad o acomodar los parámetros a sus necesidades	
<b>Alcance:</b>	
<ol style="list-style-type: none"> <li>1. El botón “Avanzado” dentro del “Generador de contraseñas” abrirá las siguientes opciones modificables: <ol style="list-style-type: none"> <li>a. Longitud.</li> <li>b. A-Z</li> </ol> </li> </ol>	

- c. a-z
  - d. 0-9
  - e. Caracteres especiales
  - f. Mínimo de caracteres numéricos
  - g. Mínimo de caracteres especiales
2. Botón “Seleccionar” que permitirá colocar la nueva contraseña segura en el formulario a completar.
  3. Botón “Cancelar” acción que descarta la contraseña presentada y cierra el modal para continuar con los demás campos del formulario.

**Observación:**

- **Longitud** (control de número): modifica la longitud de caracteres que tendrá la contraseña a crearse.
- **A-Z** (checkbox): casilla de selección para el uso de letras mayúsculas.
- **a-z** (checkbox): casilla de selección para el uso de letras minúsculas.
- **0-9** (checkbox): casilla de selección para el uso de números.
- **Caracteres especiales** (checkbox): casilla de selección para el uso de caracteres especiales.
- **Mínimo de caracteres numéricos** (control de número): modifica el número mínimo de caracteres numéricos que tendrá la contraseña a crearse.
- **Mínimo de caracteres especiales** (control de número): modifica el número mínimo de caracteres especiales que tendrá la contraseña a crearse.
- **Botón “Seleccionar”**: acción que pasara la contraseña mostrada al formulario del nuevo recurso a guardarse.
- **Botón “Cancelar”**: acción que descarta la contraseña presentada y cierra el modal para continuar con los demás campos del formulario.

**Criterios de Aceptación:**

- Se presentará un mensaje a modo de alerta cuando la contraseña sea seleccionada.

**Tabla XI Historia de usuario 010.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-010	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Modificar los recursos almacenados	
<b>Prioridad para el negocio:</b> Medio	<b>Riesgo para desarrollo:</b> Baja
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b>	
<b>Como:</b> Usuario	
<b>Quiero:</b> Administrar los recursos	
<b>Para:</b> Actualizar o eliminar los recursos guardados	
<b>Alcance:</b>	

<ol style="list-style-type: none"> <li>1. El botón “Editar” dentro de un recurso almacenado abre el formulario con los campos correspondientes según el tipo de recurso para ser actualizados.</li> <li>2. Botón “Guardar” que permitirá actualizar el recurso de la BDD.</li> <li>3. Botón “Eliminar” que permitirá eliminar el recurso de la BDD.</li> <li>4. Botón “Cancelar” que permitirá descartar los cambios realizados en el recurso.</li> </ol>
<p><b>Observación:</b></p> <ul style="list-style-type: none"> <li>• <b>Botón “Editar”:</b> acción que habilita la modificación y actualización de los diferentes campos según el tipo de recurso almacenado.</li> <li>• <b>Botón “Guardar”:</b> acción que actualizara la información en la BDD.</li> <li>• <b>Botón “Eliminar”:</b> acción que eliminar el recurso en la BDD.</li> </ul> <p><b>Botón “Cancelar”:</b> acción que descarta los cambios realizados y cierra el formulario de modificación del recurso.</p>
<p><b>Criterios de Aceptación:</b></p> <ul style="list-style-type: none"> <li>• Se presentará un mensaje a modo de alerta cuando el recurso se actualizó de forma satisfactoria.</li> </ul>

**Tabla XII Historia de usuario 011.**

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-011	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Visualizar la información sensible de los recursos previos	
<b>Prioridad para el negocio:</b> Alto	<b>Riesgo para desarrollo:</b> Medio
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<p><b>Descripción:</b></p> <p><b>Como:</b> Usuario</p> <p><b>Quiero:</b> Visualizar la información</p> <p><b>Para:</b> Copiar, autenticar o editar los recursos almacenados</p>	
<p><b>Alcance:</b></p> <ol style="list-style-type: none"> <li>1. El botón “Alternar visibilidad” en el apartado de la contraseña permite ocultar o mostrar la contraseña según el campo y tipo de recurso almacenado.</li> <li>2. El botón “Copiar” permite replicar la información del campo correspondiente según el tipo de recurso.</li> </ol>	
<p><b>Observación:</b></p> <ul style="list-style-type: none"> <li>• <b>Botón “Alternar visibilidad”:</b> acción que permite alternan entre si se muestra o no la información de la contraseña.</li> <li>• <b>Botón “Copiar”:</b> acción que permite replicar la información de los diferentes campos según el tipo de recurso como el usuario, contraseña o número de tarjeta.</li> </ul>	
<p><b>Criterios de Aceptación:</b></p> <ul style="list-style-type: none"> <li>• Se presentará un mensaje a modo de alerta cuando la información de un campo sea copiada de forma satisfactoria.</li> </ul>	



Tabla XIII Historia de usuario 012.

<b>HISTORIA DE USUARIO</b>	
<b>ID:</b> HU-012	<b>Usuario:</b> Usuario
<b>Nombre de la Historia:</b> Seleccionar el tipo de recurso que desea guardar	
<b>Prioridad para el negocio:</b> Medio	<b>Riesgo para desarrollo:</b> Medio
<b>Desarrollador responsable:</b> Michaelle Valenzuela	
<b>Descripción:</b> <b>Como:</b> Usuario <b>Quiero:</b> Seleccionar el tipo de recurso <b>Para:</b> Guardar la información acorde a las necesidades de autenticación	
<b>Alcance:</b> <ol style="list-style-type: none"> <li>1. El campo de selección de "Tipo" dentro de la creación de un nuevo recurso despliega las siguientes opciones: <ol style="list-style-type: none"> <li>a. Entrada.</li> <li>b. Tarjeta</li> <li>c. Nota segura</li> </ol> </li> </ol>	
<b>Observación:</b> <ul style="list-style-type: none"> <li>• <b>Tipo</b> (campo de selección - <i>comboBox</i>): con las opciones de los perfiles definidos (en BDD).</li> </ul>	
<b>Criterios de Aceptación:</b> <ul style="list-style-type: none"> <li>• Según la opción seleccionada el formulario del recurso a crearse cambia de manera automática.</li> </ul>	

## Producto Backlog

En la siguiente **Tabla XIV** se describe el resultado final del orden de los módulos a codificarse.

**Tabla XIV Product Backlog completo.**

<b>ELABORACIÓN DEL PRODUCT BACKLOG</b>				
<b>ID – HU</b>	<b>HISTORIA DE USUARIO</b>	<b>ITERACIÓN</b>	<b>ESTADO</b>	<b>PRIORIDAD</b>
HU001	Iniciar y cerrar sesión	1	Finalizado	Alta
HU002	Cambiar el estado de la cuenta de los usuarios	1	Finalizado	Alta
HU003	Seleccionar el tipo de encriptación de los datos	1	Finalizado	Media
HU004	Gestionar los tipos de recurso que puede guardar el usuario	2	Finalizado	Alta
HU005	Registrarse	2	Finalizado	Media
HU006	Crear una nueva entrada del recurso a autenticar	2	Finalizado	Alta
HU007	Modificar el perfil de usuario	3	Finalizado	Alta

HU008	Generar una nueva contraseña segura	3	Finalizado	Alta
HU009	Modificar los parámetros necesarios para generar una contraseña segura	3	Finalizado	Media
HU010	Modificar los recursos almacenados	4	Finalizado	Media
HU011	Visualizar la información sensible de los recursos previos	4	Finalizado	Alta
HU012	Seleccionar el tipo de recurso que desea guardar	4	Finalizado	Media

## Diagrama de casos de uso

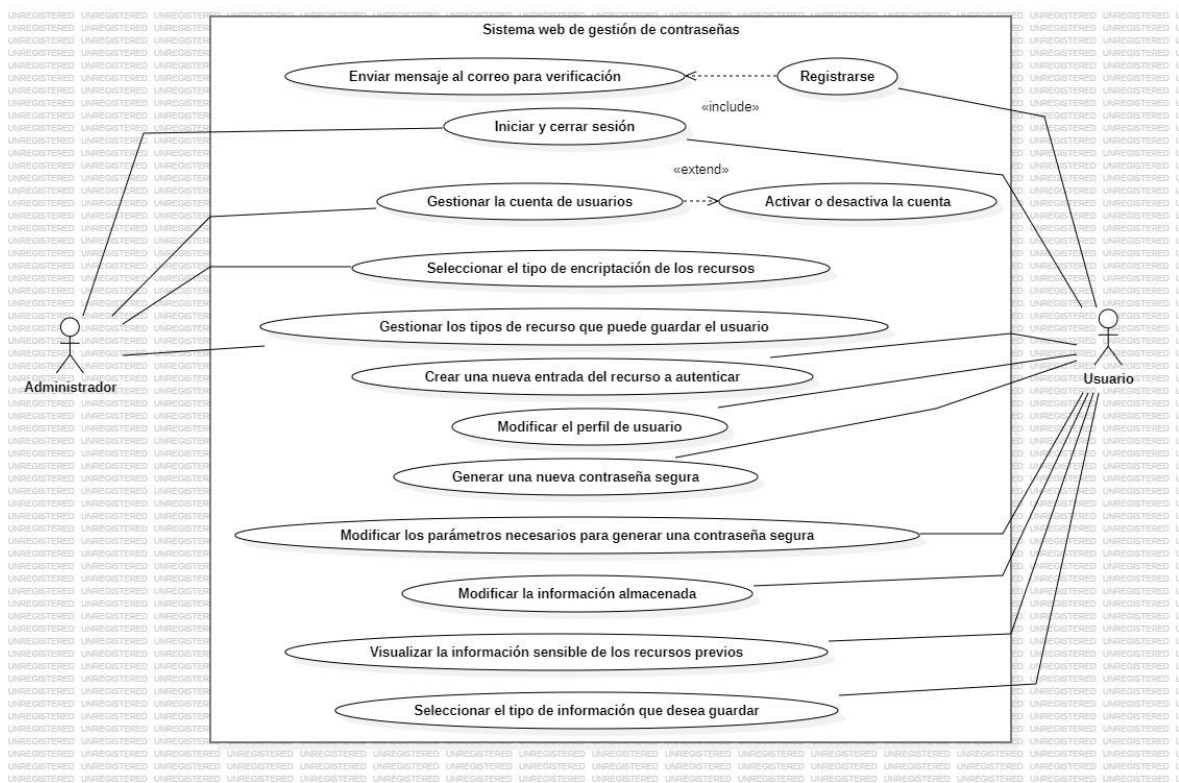


Figura I Diagrama de casos de uso del sistema.

# Diagrama de procesos

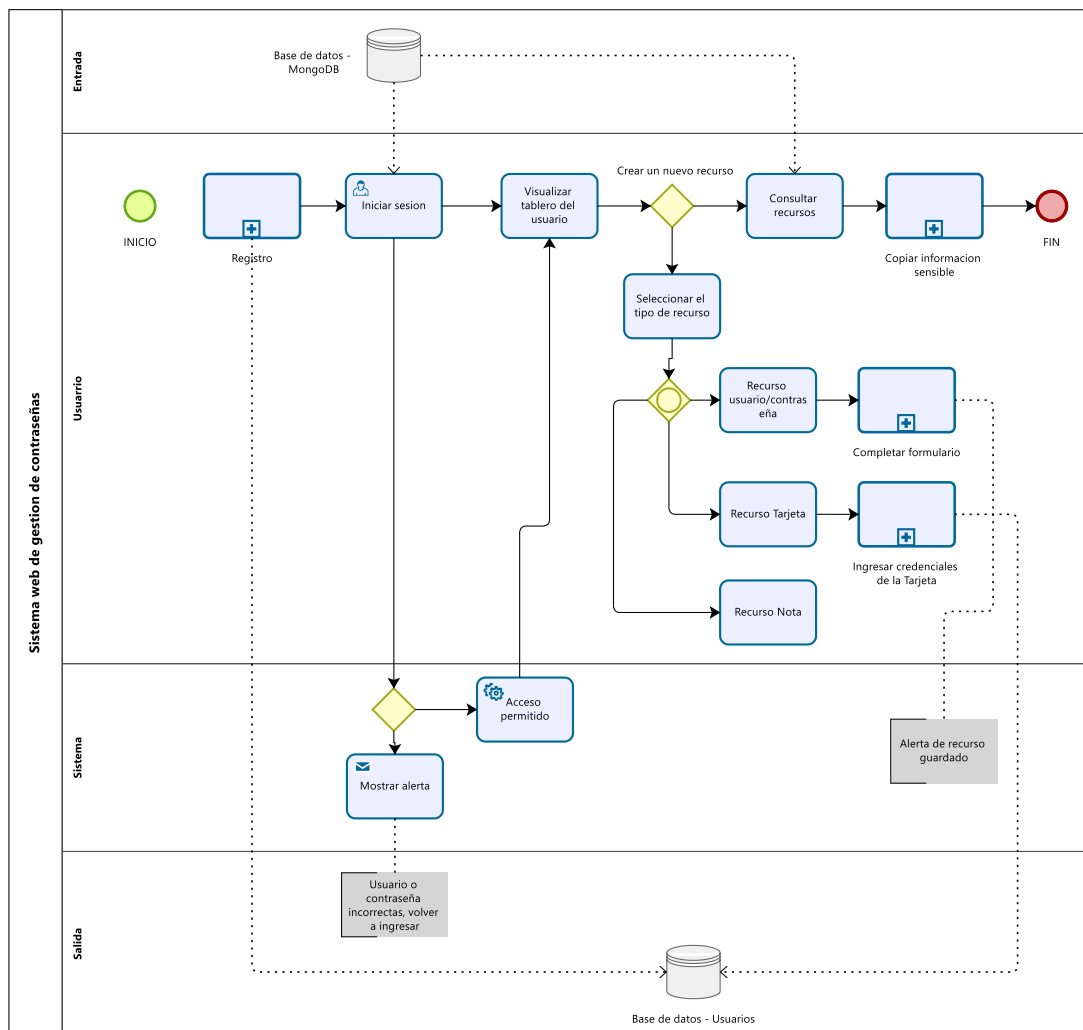
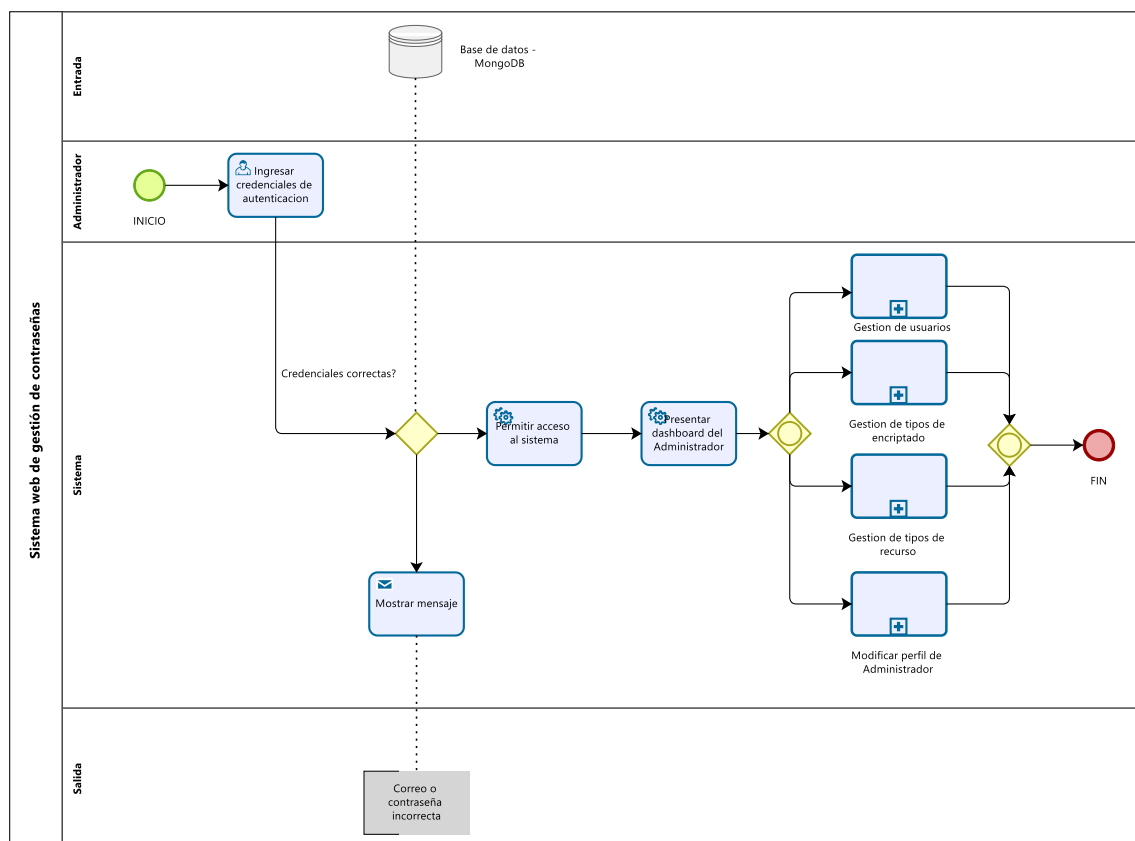


Figura II Diagrama de usuario.



**Figura III Diagrama de Administrador.**

## Sprint Backlog

En la **Tabla XV** se registra de manera completa el *Sprint Backlog* del *frontend* con cada una de las tareas a realizar.

**Tabla XV *Sprint Backlog* completo.**

ELABORACIÓN DE <i>SPRINT BACKLOG</i>						
ID – SB	NOMBRE	MÓDULO	ID-HU	HISTORIAS DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB001	Diseño e implementación del perfil administrador	Módulo – Inicio de sesión	HU001	Iniciar y cerrar sesión	<ul style="list-style-type: none"> <li>• Diseño y consumo de los <i>endpoints</i> para el inicio de sesión del administrador.</li> <li>• Validación de campos requeridos para el ingreso.</li> </ul>	40H
		Módulo – Cambio de estados de cuentas	HU002	Cambiar el estado de la cuenta de los usuarios	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para modificar el estado de las cuentas registradas de los usuarios.</li> <li>• Modificación de las cuentas en el <i>backend</i>.</li> </ul>	

		Módulo - Gestionar los tipos de información a almacenar	HU004	Gestionar los tipos de recurso que puede guardar el usuario	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> de parte del <i>backend</i> para la gestión de los tipos de información que puede almacenar el usuario.</li> <li>• Validación de campos requeridos.</li> </ul>	
SB002	Diseño e implementación del perfil usuario.	Módulo – Registro	HU005	Registrarse	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> del <i>backend</i> para el registro del usuario.</li> <li>• Validación de los campos requeridos.</li> </ul>	40H
		Módulo – Inicio de sesión	HU001	Iniciar y cerrar sesión	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para recuperar datos del <i>backend</i> para acceder al sistema.</li> <li>• Validación de los campos requeridos.</li> </ul>	



SB003	Diseño e implementación del perfil usuario.	Módulo – Nuevo recurso	HU006	Crear una nueva entrada del recurso a autenticar	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para crear una nueva entrada con información a almacenar.</li> <li>• Validación de los campos requeridos.</li> </ul>	40H
		Módulo – Modificar el perfil	HU007	Modificar el perfil de usuario	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para modificar la información proporcionada al momento de registrarse.</li> <li>• Validación de los campos requeridos.</li> </ul>	
		Módulo - Generar nueva contraseña	HU008	Generar una nueva contraseña segura	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para generar una contraseña predeterminada segura</li> <li>• Validar los campos requeridos.</li> </ul>	
		Módulo – Modificar nueva contraseña	HU009	Modificar los parámetros necesarios para generar una	<ul style="list-style-type: none"> <li>• Diseño y consumo de <i>endpoints</i> para seleccionar las opciones al</li> </ul>	

				contraseña segura	<p>momento de generar una contraseña segura.</p> <ul style="list-style-type: none"> <li>● Validar las opciones disponibles</li> </ul>	
SB004	Diseño e implementación del perfil usuario.	Módulo - Modificación de información	HU010	Modificar los recursos almacenados	<ul style="list-style-type: none"> <li>● Diseño y consumo de <i>endpoints</i> para actualizar o borrar recursos almacenados.</li> <li>● Validar nueva información ingresada</li> </ul>	40H
		Módulo – Visualización de recursos	HU011	Visualizar la información sensible de los recursos previos	<ul style="list-style-type: none"> <li>● Diseño y consumo de <i>endpoints</i> para visualizar los recursos que la fueron ingresados para su autenticación.</li> <li>● Recuperar información de la base de datos.</li> </ul>	
		Módulo – Seleccionar tipo de recurso	HU0012	Seleccionar el tipo de recurso que desea guardar	<ul style="list-style-type: none"> <li>● Diseño y consumo de <i>endpoints</i> para escoger la opción adecuada para el recurso a guardar.</li> <li>● Escribir en la base de datos</li> </ul>	

SB005	Pruebas y despliegue del sistema	<ul style="list-style-type: none"><li>● Pruebas unitarias.</li><li>● Pruebas de compatibilidad.</li><li>● Despliegue del sistema.</li></ul>	40H
	Documentación	<ul style="list-style-type: none"><li>● Trabajo de Integración Curricular.</li><li>● Anexos.</li></ul>	40H
<b>TOTAL</b>			<b>240 H</b>

## Interfaces de usuario

En esta sección se evidencian todos los prototipados del sistema *web*.



**CREE SU NUEVA CUENTA**

Ingrese su nombre

Ingrese su apellido

Ingrese su fecha de nacimiento

Ingrese su correo

Ingrese su contraseña

Repita su contraseña

**Crear cuenta**

Figura IV Prototipado del Registro del sistema.



Figura V Prototipado del panel del Administrador.



Figura VI Prototipado del módulo de Administración de cuentas.

**ADMINISTRADOR** Cerrar sesión

**Administrar cuentas**

**Seleccionar el tipo de encriptación**

**Administrar los recursos**

**Modificar perfil de administrador**

Encriptacion de contraseña TIPO ▼

Encriptacion de tarjeta TIPO ▼

Encriptacion de nota segura TIPO ▼

Detailed description: The image shows a user interface for an administrator. At the top left, the word 'ADMINISTRADOR' is displayed in bold black text. To its right is a dark blue button with the text 'Cerrar sesión'. Below this is a dark blue sidebar containing four menu items: 'Administrar cuentas', 'Seleccionar el tipo de encriptación', 'Administrar los recursos', and 'Modificar perfil de administrador'. The main content area is light blue and contains three rows of settings. Each row has a label and a dropdown menu. The first row is 'Encriptacion de contraseña' with a dropdown showing 'TIPO' and a downward arrow. The second row is 'Encriptacion de tarjeta' with a dropdown showing 'TIPO' and a downward arrow. The third row is 'Encriptacion de nota segura' with a dropdown showing 'TIPO' and a downward arrow.

Figura VII Prototipado del módulo de Selección del tipo de encriptación.

**ADMINISTRADOR** Cerrar sesión

**Administrar cuentas**

**Seleccionar el tipo de encriptación**

**Administrar los recursos**

**Modificar perfil de administrador**

Tipo de recurso A Seleccionar ▼

Tipo de recurso B Seleccionar ▼

Tipo de recurso C Seleccionar ▼

Crear nuevo recurso

Detailed description: The image shows a user interface for an administrator. At the top left, the word 'ADMINISTRADOR' is displayed in bold black text. To its right is a dark blue button with the text 'Cerrar sesión'. Below this, there is a vertical sidebar on the left with four menu items: 'Administrar cuentas', 'Seleccionar el tipo de encriptación', 'Administrar los recursos', and 'Modificar perfil de administrador'. The main content area on the right is light blue and contains three rows, each with a label 'Tipo de recurso A', 'Tipo de recurso B', and 'Tipo de recurso C' respectively, followed by a white button with a dark blue border and the text 'Seleccionar' and a downward-pointing triangle. At the bottom of this main area is a dark blue button with the text 'Crear nuevo recurso'.

Figura VIII Prototipado del módulo de Administración de recursos.



**ADMINISTRADOR** Cerrar sesión

**Administrar cuentas**

**Seleccionar el tipo de encriptación**

**Administrar los recursos**

**Modificar perfil de administrador**

**Modificar perfil**

Modificar usuario

Modificar contraseña

Repetir nueva contraseña

**Guardar**

Figura IX Prototipado del módulo de Modificar perfil – Administrador.

## Hola (Usuario)

Recurso A

Recurso B

Recurso C

Recurso D

Recurso E

Recurso F

Recurso G

### Información del recurso

Nombre

Contraseña

Usuario

**Editar**

**Borrar**

Figura X Prototipado del módulo de Visualización de los recursos.

## Hola (Usuario)

Recurso A

Recurso B

Recurso C

Recurso D

Recurso E

Recurso F

**Nuevo recurso**

### Información del nuevo recurso

Tipo

Seleccionar ▼

Nombre

Usuario

Contraseña

**Guardar**

**Cancelar**

Figura XI Prototipado del módulo de Creación de nuevo recurso.

## Hola (Usuario)

### Modificar el perfil de usuario

Usuario

Correo

Contraseña

Repita la nueva contraseña

**Guardar**




Figura XII Prototipado del módulo de modificación del perfil – Usuario.

## Hola (Usuario)

### Generador de contraseñas

zd3tSyU7

Avanzado

**Seleccionar** **Cancelar**

**Recurso**

Recurso A

Recurso B

Recurso C

Recurso D

Recurso E

Recurso F

**Nuevo recurso**

**Recurso**

Recurso A

Recurso B

Recurso C

Recurso D

Recurso E

Recurso F

**Nuevo recurso**

**Nombre**

**Usuario**

**Contraseña**

**Guardar**

**Cancelar**

Figura XIII Prototipado del módulo de Generador de contraseñas.

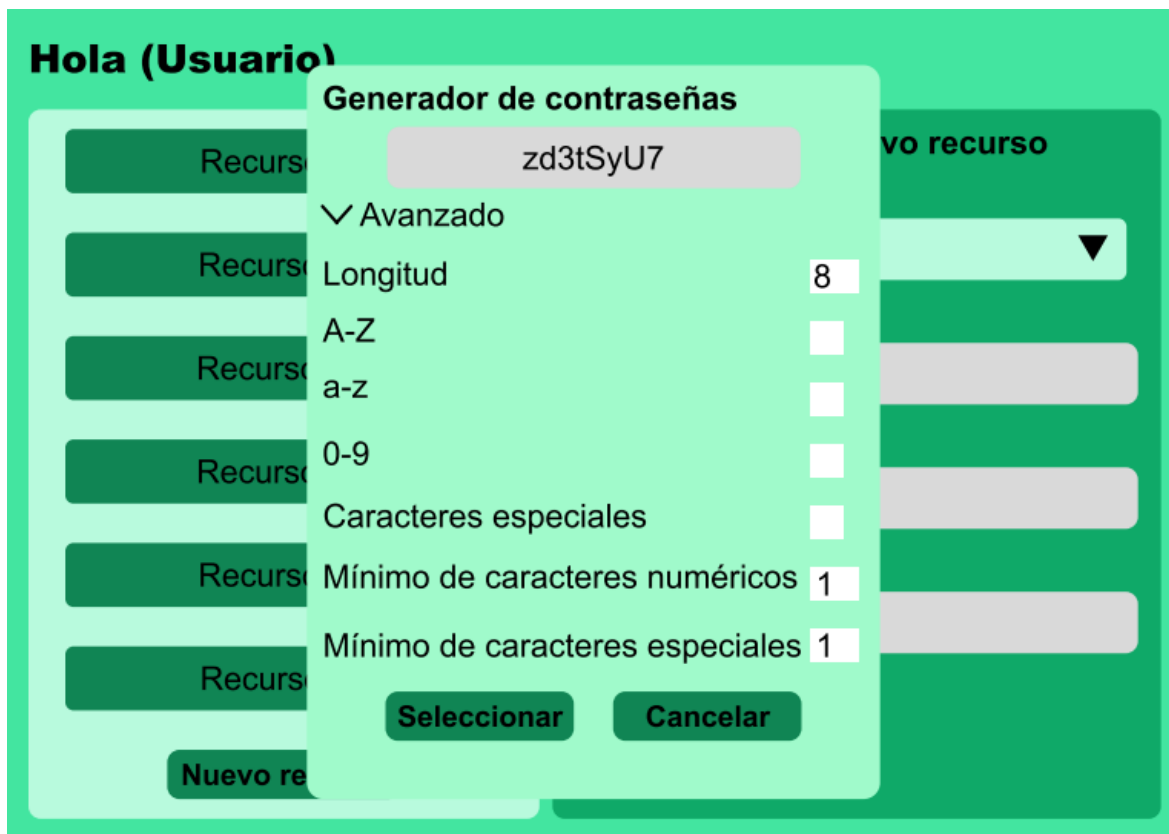


Figura XIV Prototipo del módulo de Generador de contraseñas personalizadas.

## Pruebas unitarias

En esta sección se muestran las pruebas unitarias del sistema y los resultados obtenidos de cada una de estas.

```
describe("Test Controlador-autenticacion", () => {
  testNewUser();
  testLoginAdmin();
  testAdminBlockUser();
});

describe("Test Controlador-Usuario", () => {
  testUserController();
});

describe("Test Controlador-Recursos-Usuario", () => {
  testAdminResource();
  testUserResource();
  testAdminDeleteResource();
  testAdminDeleteUser()
});
```

Figura XV Archivo *main.ts* con las pruebas a realizar.

```
// Test de registro de un nuevo usuario
it("Test crear nuevo usuario", async () => {
  const req: any = {
    body: {
      email: MailVerifier.targetUserEmail,
      name: "Test",
      lastname: "User",
      birth: "2000-01-20",
      password: process.env.APP_SERVER_TEST_USER_PASSWORD,
      confirm_password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await registerUser(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura XVI Test de registro de nuevo usuario.

```
// Test de confirmación de cuenta
it("Test confirmar cuenta", async () => {
  const res = await MailVerifier.waitForEmail();
  expect(res).toBe(true);
}, 250000);
});
```

Figura XVII Test de confirmación de cuenta de usuario.

```
it("Test inicio de sesion [ADMIN]", async () => {
  const req: ReqLoginType = {
    body: {
      email: process.env.APP_SERVER_TEST_USER_MAIL,
      password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await loginUser(req, res);
  expect(res.status.mock.calls).toEqual([[201]]);
  expect((res.json.mock.lastCall[0] as any).data.token_auth).toBeDefined();
  TOKEN = (res.json.mock.lastCall[0] as any).data.token_auth;
}, 30000);
```

Figura XVIII Test de cierre de sesión.

```

// Funcion que inicia sesion con un usuario
const loginTestUser = async (validateBan: boolean) => {
  const req: ReqLoginType = {
    body: {
      email: MailVerifier.targetUserEmail,
      password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await loginUser(req, res);
  if (validateBan) {
    expect(res.status.mock.calls).toEqual([[404]]);
    expect((res.json.mock.lastCall[0] as any).msg).toEqual(
      "Esta cuenta ha sido deshabilitada por el Admin"
    );
  } else {
    expect(res.status.mock.calls).toEqual([[201]]);
  }
};

```

Figura XIX Función de inicio de sesión de usuario de prueba.

```

// Funcion que cambia el estado de la cuenta del usuario
const changeUserState = async (userIsActive: boolean) => {
  const req: ReqResourceType = {
    body: {
      userIsActiveByAdmin: userIsActive
    },
    params: {
      id: TEST_USER_ID
    }
  };
  await deactivateUser(req, res);
  expect(res.status.mock.calls).toEqual([[201]]);
};

```

Figura XX Función de cambio de estado de cuenta de usuario.

```

it(
  "Test de inicio de sesion usuario verificado",
  async () => loginTestUser(false),
  30000
);

// Test de cambio de estado de un usuario
it(
  "Test cambio de estado cuenta [ADMIN] inactivo",
  async () => changeUserState(false),
  30000
);

// Test de inicio de sesión de un usuario
it(
  "Test de inicio de sesion usuario inactivo",
  async () => loginTestUser(true),
  30000
);

```

Figura XXI Bloque de pruebas de modificación de estado de cuentas.

```

// Test de cambio de estado de un usuario
it(
  "Test cambio de estado cuenta [ADMIN] activo",
  async () => changeUserState(true),
  30000
);

// Test de inicio de sesión de un usuario
it(
  "Test de inicio de sesion usuario activo",
  async () => loginTestUser(false),
  30000
);
);

```

Figura XXII Bloque de pruebas de modificación de estado de cuentas.



```

//Test de modificacion de perfil de usuario
const req: ReqEditProfileType = {
  body: {
    email: MailVerifier.targetUserEmail,
    password: rollback
      ? process.env.APP_SERVER_TEST_USER_PASSWORD
      : "Nuevapassword123!",
    confirm_password: rollback
      ? process.env.APP_SERVER_TEST_USER_PASSWORD
      : "Nuevapassword123!"
  },
  headers: {
    usuario_autorizacion: TOKEN
  },
  params: {
    id: USER_ID._id
  }
};
await editProfileUser(req, res);
expect(res.status.mock.calls).toEqual([[201]]);

```

Figura XXIII Test de modificación de perfil de usuario.

```

// Test de inicio de sesión de usuario
it("Inicio de sesion", async () => {
  const req: ReqLoginType = {
    body: {
      email: MailVerifier.targetUserEmail,
      password: process.env.APP_SERVER_TEST_USER_PASSWORD
    }
  };
  await loginUser(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
  expect((res.json.mock.lastCall[0] as any).data.token_auth).toBeDefined();
  TOKEN = (res.json.mock.lastCall[0] as any).data.token_auth;
  USER_ID = decodeTokenUser(
    (res.json.mock.lastCall[0] as any).data.token_auth
  ) as JwtPayload;
}, 30000);

```

Figura XXIV Test de inicio de sesión de usuario.

```

const bodyCreateResourceAdmin: ReqCreateResourceType["body"] = {
  fields: [
    {
      _id: generateRandomString(),
      label_input: "Primer campo de prueba",
      type_input: "text"
    },
    {
      _id: generateRandomString(),
      label_input: "Segundo campo de prueba",
      type_input: "password"
    }
  ],
  id_user_ref: undefined,
  name_resource: "Recurso de prueba",
  name_type_encrypt: "JWT"
};

```

Figura XXV Cuerpo de crear nuevo recurso.

```

// Test de creacion de un nuevo tiop recurso
it("Crear nuevo tipo de recurso", async () => {
  bodyCreateResourceAdmin.id_user_ref = USER_ID._id;
  const req: ReqCreateResourceType = {
    body: bodyCreateResourceAdmin,
    headers: {
      usuario_autorizacion: TOKEN
    }
  };
  await adminCreateResource(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);

```

Figura XXVI Test de creación de un nuevo recurso.

```
//Test de cambio del tipo de encriptacion
it("Test de cambio del tipo de encriptado", async () => {
  const req: ReqCreateResourceType = {
    body: {
      id_user_ref: USER_ID._id,
      name_type_encrypt: "BASE 64"
    },
    headers: {
      usuario_autorizacion: TOKEN
    },
    params: {
      id: RESOURCE_ID
    }
  };
  await adminChangeTypeEncryptResource(req, res);
  // Verifica que el status de la respuesta sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura XXVII Test de modificación del tipo de encriptación.

```
// Test de creación de un nuevo recurso
it("Test crear nuevo recurso [USER]", async () => {
  const req: ReqCreateResourceType = {
    body: {
      id_resource_admin: RESOURCE_ID,
      id_user_ref: USER_ID._id,
      inputs_data: [
        {
          nombre_del_recurso: "Recurso de prueba User",
          [bodyCreateResourceAdmin.fields[0].label_input]:
            "First test field",
          [bodyCreateResourceAdmin.fields[1].label_input]:
            "Second test field"
        }
      ],
      name_resource_admin: bodyCreateResourceAdmin.name_resource
    },
    headers: {
      usuario_autorizacion: TOKEN
    }
  };
  await userCreateResource(req, res);
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura XXVIII Test de creación de un nuevo recurso.

```
// Test de eliminacion de recurso
it("Test de eliminar recurso", async () => {
  const req: ReqCreateResourceType = {
    body: {
      id_user_ref: USER_ID._id
    },
    headers: {
      usuario_authorized: TOKEN
    },
    params: {
      id: RESOURCE_ID
    }
  };
  await adminDeleteResource(req, res);
  // Se espera que el estado sea 201
  expect(res.status.mock.calls).toEqual([[201]]);
}, 30000);
```

Figura XXIX Test de eliminar recurso de prueba.

```

PASS test/main.test.ts (84.054 s)
Test Controlador-autenticacion
  Test de registro de nuevo usuario
    ✓ Test crear nuevo usuario (291 ms)
    ✓ Test confirmar cuenta (69631 ms)
  Test Inicio de sesion Administrador
    ✓ Test inicio de sesion [ADMIN] (148 ms)
    ✓ Test cierre de sesion [ADMIN] (12 ms)
  Test Controlador Usuario-Administrador
    ✓ Test de busqueda de usuario [ADMIN] (19 ms)
    ✓ Test de inicio de sesion usuario verificado (113 ms)
    ✓ Test cambio de estado cuenta [ADMIN] inactivo (53 ms)
    ✓ Test de inicio de sesion usuario inactivo (11 ms)
    ✓ Test cambio de estado cuenta [ADMIN] activo (35 ms)
    ✓ Test de inicio de sesion usuario activo (115 ms)
  Test Controlador-Usuario
    Perfil de usuario [USER]
      ✓ Inicio de sesion (116 ms)
      ✓ Editar perfil - Cambiar contraseña (124 ms)
      ✓ Editar perfil - Repetir contraseña (118 ms)
    Test Controlador-Recursos-Usuario
      Tipo de recurso [ADMIN]
        ✓ Inicio de sesion [ADMIN] (120 ms)
        ✓ Validar si el recurso no existe (42 ms)
        ✓ Crear nuevo tipo de recurso (43 ms)
        ✓ Validar si el recurso existe (24 ms)
        ✓ Test de cambio del tipo de encriptado (44 ms)
        ✓ Validar si el recurso cambio de encriptacion (21 ms)
      Recurso de usuario[USER]
        ✓ Inicio de sesion (116 ms)
        ✓ Test crear nuevo recurso [USER] (25 ms)
      Borrar Tipo de recurso [ADMIN]
        ✓ Inicio de sesion (115 ms)
        ✓ Test de eliminar recurso (30 ms)
    Test Eliminar Controlador Usuario-Administrador
      ✓ Test eliminar usuario de prueba [ADMIN] (10 ms)

Test Suites: 1 passed, 1 total
Tests:      24 passed, 24 total
Snapshots: 0 total
Time:      84.395 s

```

Figura XXX Resultados de las pruebas unitarias.

```

//Test del generador de contraseñas
test('Genera una nueva contraseña', async () => {
  const wrapper = mount(GeneratePassword)
  wrapper.vm.onGenerateNow()
  // verifica que el password se genero correctamente
  expect(wrapper.vm.password_generate).toBeTruthy()
})
// Test para verificar que se generen contraseñas diferentes
test('Genera una contraseña diferente', async () => {
  const wrapper = mount(GeneratePassword)
  wrapper.vm.onGenerateNow()
  // verifica que el password se genero correctamente
  const firstPassword = wrapper.vm.password_generate
  expect(firstPassword).toBeTruthy()
  wrapper.vm.onGenerateNow()
  const secondPassword = wrapper.vm.password_generate
  // verifica que el password no sea igual al anterior
  expect(secondPassword).not.toBe(firstPassword)
})

```

Figura XXXI Test del generador de contraseña.

```

// Test para verificar que se generen contraseñas con letras
test('Genera una contraseña con letras y longitud de 8 caracteres', async () => {
  const wrapper = mount(GeneratePassword)
  const obj = {
    target: {
      name: 'checkbox1',
      checked: true
    }
  }
  // cambia la longitud de la prop password
  wrapper.vm.longitud = 8
  wrapper.vm.onOptionalGeneratorPass(obj)
  wrapper.vm.onOptionalGenerateNow(obj)
  const password = wrapper.vm.password_generate
  expect(password.length).toBe(8)
})

```

Figura XXXII Test de la modificación del apartado de letras.

```

// Test para verificar que se generen contraseñas con números
test('Genera una contraseña con numeros y logitus de 10 caracteres', async () => {
  const wrapper = mount(GeneratePassword)
  const obj = {
    target: {
      name: 'checkbox3',
      checked: true
    }
  }
  // cambia la longitud de la prop password
  wrapper.vm.longitud = 10
  wrapper.vm.onOptionalGeneratorPass(obj)
  wrapper.vm.onOptionalGenerateNow(obj)
  const password = wrapper.vm.password_generate
  expect(password.length).toBe(10)
  expect(password).toMatch(/^[0-9]+$/)
})

```

Figura XXXIII Test de generación de contraseña con números y longitud de 10 caracteres.

```

✓ test/main.test.ts (6) 330ms
  ✓ Generador de contraseñas (5)
    ✓ Renderizado del componentes
    ✓ Genera una nueva contraseña
    ✓ Genera una contraseña diferente
    ✓ Genera una contraseña con letras y longitud de 8 caracteres
    ✓ Genera una contraseña con numeros y logitus de 10 caracteres
  ✓ -- (1)
    ✓ El componente se monta correctamente

Test Files  1 passed (1)
  Tests     6 passed (6)
  Start at  22:42:30
  Duration  1.12s

PASS  Waiting for file changes...
      press h to show help, press q to quit

```

Figura XXXIV Resultado de las pruebas unitarias del *frontend*.



```

it('Seleccionar el tipo de recurso que desea guardar', () => {
  cy.get('a[href="/entries"]', {
    timeout: 5000
  }).click()
  cy.url().should('include', '/entries')
  cy.get('div.dash_content_main > div:nth-child(2) > div > button').click()
  // Selecciona el tipo de recurso que desea guardar
  cy.get('div.relative.z-10 > div.fixed.inset-0.z-10.w-screen.overflow-y-auto div > select').select(
    'Tipo de tarjeta')
  // Selecciona el campo de nombre del recurso y escribe Prueba
  cy.get('input[id="nombre_del_recurso_0"]').type('Prueba')
  cy.get('input[id="nombre_tarjeta_1"]').type('Prueba tarjeta')
  cy.get('input[id="nrodetarjeta_2"]').type('112233445566')
  cy.get('input[id="mesdeexpiracion_3"]').type('05')
  cy.get('input[id="anideexpiracion_4"]').type('25')
  cy.get('input[id="codigodeseguridad_5"]').type('123')
  cy.get('div.relative.z-10 div:nth-child(1) > button').click()
  cy.get('div.LAYOUT_APP_SIDE_RIGHT > div.dash_content_main > div > div')
    .find('.msg_no_data_stored')
    .should('not.exist');
})

```

Figura XXXV Test de selección del tipo de recurso a crearse.

```

it('Visualizar la información sensible de los recursos previos', () => {
  cy.get('a[href="/entries"]', {
    timeout: 5000
  }).click()
  cy.url().should('include', '/entries')
  // Verifica que exista un recurso guardado con el nombre Prueba
  cy.get('div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)').invoke('text').then((text)
=> {
  expect(text).to.equal('Prueba');
})
  cy.get('div.dash_content_main > div > div > div table div > button.BTN_WARNING_1').click()
  validateType('input[id="nrodetarjeta_2"]', 'password')
  cy.get('div[id="nrodetarjeta"]').click()
  validateType('input[id="nrodetarjeta_2"]', 'text')
  validateType('input[id="codigodeseguridad_5"]', 'password')
  cy.get('div[id="codigodeseguridad"]').click()
  cy.get('div[id="codigodeseguridad"]').click()
  validateType('input[id="codigodeseguridad_5"]', 'text')
})

```

Figura XXXVI Test de visualizar los recursos almacenados.

```

it('Eliminar los recursos previos guardados', () => {
  cy.get('a[href="/entries"]', {
    timeout: 5000
  }).click()
  cy.url().should('include', '/entries')
  // Verifica que exista un recurso guardado con el nombre Modificación de recurso
  cy.get('div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)').invoke('text').then((text)
=> {
  expect(text).to.equal('Modificación de recurso');
})
  // Hace click en el botón de eliminar recurso
  cy.get('div.dash_content_main > div > div > div table div > button.BTN_DANGER_1').click()
  cy.get('div.p-component-overlay-enter > div button.p-confirm-dialog-accept.p-button-danger').click()
  cy.get('div.LAYOUT_APP_SIDE_RIGHT > div.dash_content_main > div > div')
    .find('.msg_no_data_stored')
    .should('exist');
})

```

Figura XXXVII Test de eliminar recurso guardado.

```

(xhr) ● POST 201 http://localhost:7000/user/create-resource
get div.LAYOUT_APP_SIDE_RIGHT > div.dash_content_main > div > div
find .msg_no_data_stored
-assert expected .msg_no_data_stored not to exist in the DOM
(xhr) ● GET 201 http://localhost:7000/user/get-resources

```

Figura XXXVIII Resultado de la selección del tipo de recurso.

```
✓ Visualizar la información sensible de los recursos previos
> BEFORE EACH
▼ TEST BODY
1  get a[href="/entries"]
   (xhr) ● POST 201 http://localhost:7000/auth/login
   (page load) --page loaded--
   (new url) http://localhost:5173/dashboard
2  -click
   (new url) http://localhost:5173/entries
3  url
4  -assert expected http://localhost:5173/entries to include /entries
5  get div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)
   (xhr) ● GET 201 http://localhost:7000/user/get-resources
6  invoke .text()
7  -assert expected Prueba to equal Prueba
8  get div.dash_content_main > div > div > div table div > button.BTN_WARNING_1
9  -click
   (xhr) ● GET 201 http://localhost:7000/admin/get-resources
10 get input[id="nrodetarjeta_2"]
```

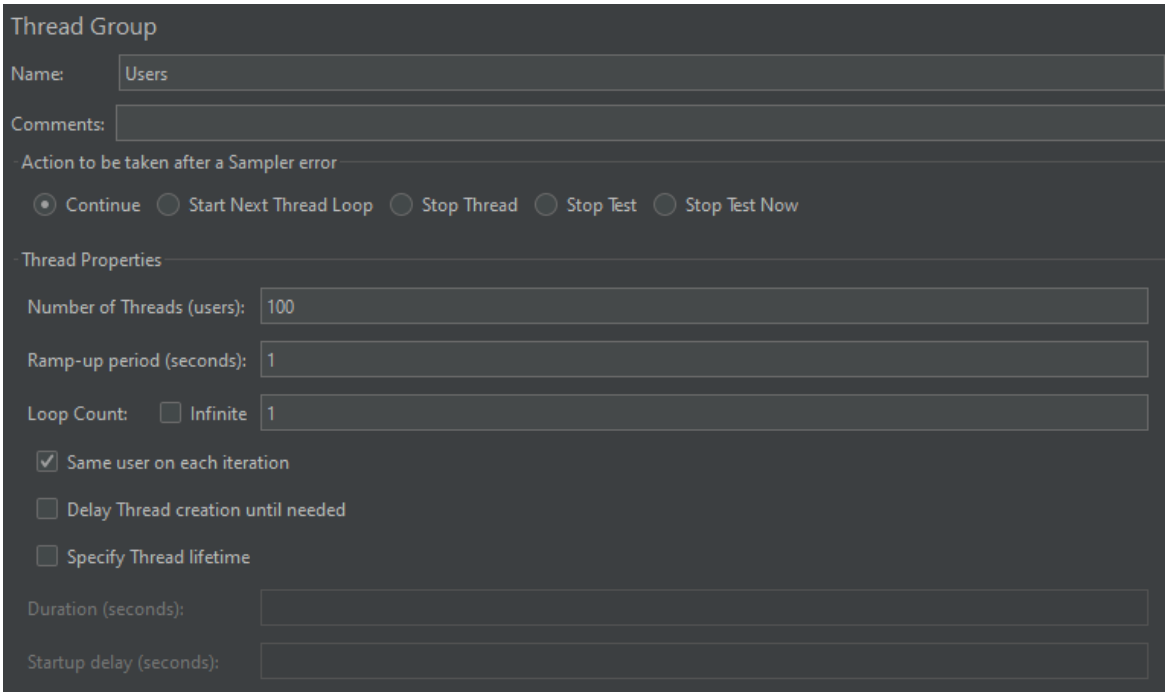
Figura XXXIX Resultado de la prueba de visualización de los recursos.

```
✓ Modificar los recursos almacenados
> BEFORE EACH
✓ TEST BODY
1  get a[href="/entries"]
   (xhr) ● POST 201 http://localhost:7000/auth/login
   (page load) --page loaded--
   (new url) http://localhost:5173/dashboard
2  -click
   (new url) http://localhost:5173/entries
3  url
4  -assert expected http://localhost:5173/entries to include /entries
5  get div.dash_content_main table > tr:nth-child(2) > td:nth-child(1)
   (xhr) ● GET 201 http://localhost:7000/user/get-resources
6  invoke .text()
7  -assert expected Prueba to equal Prueba
8  get div.dash_content_main > div > div > div table div > button.BTN_WARNING_1
9  -click
   (xhr) ● GET 201 http://localhost:7000/admin/get-resources
10 get input[id="nombre_del_recurso_0"]
    (xhr) ● GET 201 http://localhost:7000/admin/get-resource/65b4108ec6a6f5c03bf8afd2
```

Figura XL Resultado de la prueba de modificar los recursos almacenados.

## Pruebas de carga y resultados

La herramienta de Jmeter al contar con un apartado grafico las configuraciones son directamente desde el aplicativo.



The image shows the configuration window for a Thread Group in JMeter. The window is titled "Thread Group" and contains the following fields and options:

- Name:** A text input field containing "Users".
- Comments:** An empty text input field.
- Action to be taken after a Sampler error:** A group of radio buttons with the following options:  Continue,  Start Next Thread Loop,  Stop Thread,  Stop Test, and  Stop Test Now.
- Thread Properties:** A section containing several fields:
  - Number of Threads (users):** A text input field containing "100".
  - Ramp-up period (seconds):** A text input field containing "1".
  - Loop Count:** A checkbox labeled "Infinite" (unchecked) followed by a text input field containing "1".
  - Same user on each iteration
  - Delay Thread creation until needed
  - Specify Thread lifetime
- Duration (seconds):** An empty text input field.
- Startup delay (seconds):** An empty text input field.

**Figura XLI Parámetros de la prueba de estrés.**

HTTP Request

Name:

Comments:

Basic Advanced

Web Server

Protocol [http]:  Server Name or IP:  Port Number:

HTTP Request

POST Path:  Content encoding:

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

```

1 {
2   "email": "${CORREO}",
3   "password": "${PASSWORD}"
4 }

```

**Figura XLII** Parámetros de la solicitud de inicio de sesión.

Web Server

Protocol [http]:  Server Name or IP:  Port Number:

HTTP Request

POST Path:  Content encoding:

Redirect Automatically  Follow Redirects  Use KeepAlive  Use multipart/form-data  Browser-compatible headers

Parameters Body Data Files Upload

```

1 {
2   "id_resource_admin": "${RESOURCE_ID}",
3   "id_user_ref": "${USER_ID}",
4   "inputs_data": {
5     "Nombre del recurso": "Primer campo",
6     "Descripción": "Segundo campo"
7   },
8   "name_resource_admin": "${RESOURCE_NAME}"
9 }

```

**Figura XLIII** Parámetros de la creación de un nuevo recurso.

User Defined Variables

Name:

Comments:

User Defined Variables		
Name:	Value	Description
RESOURCE_NAME	Nota	El nombre del recurso
RESOURCE_ID	65c0b61a247f26fb1b94d035	El ID del recurso
USER_ID	65c0b37c247f26fb1b94cf81	El ID del usuario
CORREO	michaelle1.1llumiquinga@gmail.com	El correo del usuario
PASSWORD	qawseD5@	La contraseña del usuario

**Figura XLIV Definición de las variables de entorno.**

View Results in Table

Name:

Comments:

Write results to file / Read from file

Filename   Log/Display Only:  Errors  Successes

Sample #	Start Time	Thread Name	Label	Sample Time(...)	Status	Bytes	Sent Bytes	Latency	Connect Time(...)
1	00:16:13.056	Users 2-22	HTTP Request ...	1064	✓	1076	294	1064	237
2	00:16:13.226	Users 2-4	HTTP Request ...	951	✓	1076	294	951	67
3	00:16:13.046	Users 2-41	HTTP Request ...	1043	✓	1076	294	1030	248
4	00:16:13.199	Users 2-57	HTTP Request ...	1184	✓	1076	294	1184	98
5	00:16:13.179	Users 2-85	HTTP Request ...	1244	✓	1076	294	1244	117
6	00:16:13.081	Users 2-74	HTTP Request ...	1378	✓	1076	294	1378	217
7	00:16:13.175	Users 2-48	HTTP Request ...	1300	✓	1076	294	1300	124
8	00:16:13.063	Users 2-50	HTTP Request ...	1430	✓	1076	294	1430	236
9	00:16:13.172	Users 2-79	HTTP Request ...	1347	✓	1076	294	1347	127
10	00:16:13.046	Users 2-11	HTTP Request ...	1497	✓	1076	294	1497	254
11	00:16:13.187	Users 2-77	HTTP Request ...	1368	✓	1076	294	1368	115
12	00:16:13.048	Users 2-83	HTTP Request ...	1643	✓	1076	294	1643	254
13	00:16:13.066	Users 2-89	HTTP Request ...	1641	✓	1076	294	1641	236
14	00:16:13.028	Users 2-40	HTTP Request ...	1713	✓	1076	294	1713	278
15	00:16:13.036	Users 2-65	HTTP Request ...	1747	✓	1076	294	1747	267
16	00:16:13.036	Users 2-47	HTTP Request ...	1766	✓	1076	294	1765	267
17	00:16:13.186	Users 2-26	HTTP Request ...	1641	✓	1076	294	1641	108
18	00:16:13.062	Users 2-44	HTTP Request ...	1872	✓	1076	294	1872	235
19	00:16:13.017	Users 2-100	HTTP Request ...	1946	✓	1076	294	1946	291
20	00:16:13.041	Users 2-59	HTTP Request ...	1939	✓	1076	294	1939	266
21	00:16:13.038	Users 2-70	HTTP Reaquest ...	1972	✓	1076	294	1972	272

**Figura XLV Resultados de la prueba de carga.**

## Pruebas funcionales

Para la documentación de esta prueba es necesario apoyarnos de un documento externo al de este proyecto con los parámetros y descripciones necesarios.

N°	Reportado por: (Persona)	Aplicación	Rol	Componente	Nivel (alto, medio, bajo)	Incidente	Descripción	Recomendación	*****	*****	*****	*****	*****
1	Desarrollador	Sistema Web	Administrador y Usuario	Iniciar y cerrar sesión	alto	Diseño	La pantalla de inicio de sesión no se renderiza	Revisar el routing del inicio de sesión	NO	OK	OK	OK	OK
2	Desarrollador	Sistema Web	Administrador	Cambiar el estado de la cuenta de los usuarios	medio	Funcionalidad	No se actualiza el cambio de estado en la BDD y no se notifica el cambio por correo.	Modificar el servicio de correo de Google cloud y actualizar el token	NO	OK	OK	OK	OK
3	Desarrollador	Sistema Web	Administrador	Seleccionar el tipo de encriptación de los datos	alto	Funcionalidad	No hay cambio de encriptación	Emplear otra librería de encriptación	NO	NO	OK	OK	OK
4	Desarrollador	Sistema Web	Administrador	Gestionar los tipos de recursos que puede guardar el usuario	medio	Funcionalidad	No se reinicia la creación de un nuevo recurso	Reiniciar el arreglo del la creación de un nuevo recurso	NO	NO	OK	OK	OK
5	Desarrollador	Sistema Web	Usuario	Registrarse	alto	Funcionalidad	No se envía el correo de verificación	Revisar el servicio de Google cloud. Actualizar el token de google	NO	NO	OK	OK	OK
6	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso	alto	Funcionalidad	No se crea el nuevo recurso	Revisar la funcionalidad del envío de datos al backend	NO	OK	OK	OK	OK
7	Desarrollador	Sistema Web	Usuario	Modificar el perfil de usuario	alto	Funcionalidad	No se actualiza la información del correo y la contraseña	Revisar la funcionalidad de actualización del correo	NO	NO	OK	OK	OK
8	Desarrollador	Sistema Web	Usuario	Generar una nueva contraseña segura	alto	Funcionalidad	Contraseña predeterminada muy pequeña	Revisar el componente del generador de contraseñas en el frontend	NO	OK	OK	OK	OK
9	Desarrollador	Sistema Web	Usuario	Modificar los parámetros necesarios para generar una contraseña	medio	Funcionalidad	No se cumple con los parámetros de la contraseña	Modificar los campos y las opciones en el template del componente	NO	OK	OK	OK	OK
10	Desarrollador	Sistema Web	Usuario	Modificar los recursos almacenados	medio	Funcionalidad	No se actualiza los campos modificados	Revisar la funcionalidad de actualización del los recursos	NO	OK	OK	OK	OK
11	Desarrollador	Sistema Web	Usuario	Visualizar la información sensible de los recursos	alto	Diseño	No se renderiza el componente de los recursos almacenados	Revisar el componente del frontend al momento de montarlo	NO	NO	NO	OK	OK
11	Desarrollador	Sistema Web	Usuario	Seleccionar el tipo de recurso que	medio	Diseño	No se renderiza los nuevos tipos de recurso	Revisar el estado del backend en el frontend	NO	NO	NO	OK	OK

Figura XLVI Matriz de pruebas funcionales.

N°	Reportado por:	Aplicación	Rol	Componente	Nivel (alto, medio, bajo)	Incidente	Descripción	Recomendación	Solventado
1	Desarrollador	Sistema Web	Administrador	Gestionar los tipos de recursos que puede guardar el usuario	medio	Funcionalidad	No se reinicia la creación de un nuevo recurso	Reiniciar el arreglo de la creación de un nuevo recurso	si
2	Desarrollador	Sistema Web	Usuario	Registrarse	alto	Funcionalidad	No se envía el correo de verificación	Revisar el servicio de Google cloud. Actualizar el token de google	si
3	Desarrollador	Sistema Web	Usuario	Modificar el perfil de usuario	alto	Funcionalidad	No se actualiza la información del correo y la contraseña	Revisar la funcionalidad de actualización del correo	si
4	Desarrollador	Sistema Web	Usuario	Registrarse	alto	Funcionalidad	No se visualiza la ventana de verificación de cuenta	Revisar el componente de verificación de cuenta o revisar envío de correo en backend	si
5	Desarrollador	Sistema Web	Usuario	Modificar el perfil de usuario	alto	Funcionalidad	No se visualiza la ventana de recuperar contraseña	Revisar el componente del frontend de recuperar cuenta o archivo de enrutado	si
6	Desarrollador	Sistema Web	Usuario	Registrarse	bajo	Diseño	Notificación de registro con error de tipo	Modificar la notificación en el componente de registro	si
7	Desarrollador	Sistema Web	Usuario	Modificar el perfil de usuario	medio	Funcionalidad	Campo de correo en la modificación de perfil permite ingreso de correos no validos	Revisar el componente de registro en el frontend	si
8	Desarrollador	Sistema Web	Usuario	Registrarse	medio	Funcionalidad	Campo de correo en el registro permite ingreso de correos no validos	Revisar el componente de modificar perfil en el frontend	si

Figura XLVII Pruebas funcionales del Sprint 1.



Nº	Reportado por:	Aplicación	Rol	Componente	Nivel (alto, medio, bajo)	Incidente	Descripción	Recomendación	Solventado
1	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso a autenticar	alto	Funcionalidad	No se crea el nuevo recurso	Revisar la funcionalidad del envío de datos al backend	si
2	Desarrollador	Sistema Web	Usuario	Generar una nueva contraseña segura	alto	Funcionalidad	Contraseña predeterminada muy pequeña	Revisar el componente del generador de contraseñas en el frontend	si
3	Desarrollador	Sistema Web	Usuario	Modificar los parámetros necesarios para generar una contraseña segura	medio	Funcionalidad	No se cumple con los parametros de la contraseña	Modificar los campos y las opciones en el template del componente	si
4	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso a autenticar	bajo	Diseño	Modificar el label de Nombre tarjeta y Nro tarjeta	Modificar los labels en el componente del recurso en el backend	si
5	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso a autenticar	medio	Funcionalidad	Mes de expiracion de tarjeta permitir solo los numeros de 1-12	Modificar el input del recurso en el frontend	si
6	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso a autenticar	medio	Funcionalidad	Se permite el guardar tarjetas caducadas	Codificar la funcionalidad de verificar la fecha de tarjetas ingresadas	si
7	Desarrollador	Sistema Web	Usuario	Crear una nueva entrada del recurso a autenticar	medio	Funcionalidad	Aumentar el numero de caracteres permitidos en el campo de descripcion	Modificar los valores del input del recurso nota	si
8	Desarrollador	Sistema Web	Usuario	Modificar los parámetros necesarios para generar una contraseña segura	alto	Funcionalidad	Los checkbox del generador de contraseñas no funcionan adecuadamente	Revisar el componente del generador de contraseñas en el frontend	si

**Figura XLVIII Pruebas funcionales del Sprint 2.**

Nº	Reportado por:	Aplicación	Rol	Componente	Nivel (alto, medio, bajo)	Incidente	Descripción	Recomendación	Solventado
1	Desarrollador	Sistema Web	Usuario	Modificar los recursos almacenados	medio	Funcionalidad	No se actualiza los campos modificados	Revisar la funcionalidad de actualizacion de los recursos	si
2	Desarrollador	Sistema Web	Usuario	Visualizar la información sensible de los recursos previos	alto	Diseño	No se renderiza el componente de los recursos almacenados	Revisar el componente del frontend al momento de montarlo	si
3	Desarrollador	Sistema Web	Usuario	Seleccionar el tipo de recurso a guardar	medio	Diseño	No se renderiza los nuevos tipos de recurso	Revisar el estado del backend en el frontend	si
4	Desarrollador	Sistema Web	Usuario	Visualizar la información sensible de los recursos previos	bajo	Diseño	Retirar la columna del "encriptado" en el tablero del usuario	Modificar el componente del tablero del usuario en el frontend	si

**Figura XLIX Pruebas funcionales del Sprint 3.**

### **ANEXO III**

Se adjunta el enlace con el manual de usuario el cual es un video explicando el uso del sistema, el manual de instalación es un documento con la instalación del sistema.

Manual de usuario

<https://youtu.be/2VrDlvs25TM>

Manual de instalación

<https://drive.google.com/file/d/1KtsHnyhWRrYOEvpj4TwXND0ULAz60ti5/view?usp=sharing>

## ANEXO IV

Por último, se describen las credenciales de acceso al sistema de todos los roles.

Credenciales de usuario y administrador

<b>ROL</b>	<b>CORREO</b>	<b>CONTRASEÑA</b>
Administrador	kosanot162@lendfash.com	qawseD5@
Usuario	hokowa6058@lendfash.com	qawseD5@q

El enlace del sistema:

<https://passwordfront.vercel.app>