

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**DESARROLLO DE UN KEYLOGGER PARA WINDOWS 11 CON UN
DASHBOARD WEB**

DESARROLLO DE UN DASHBOARD WEB

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

ALEXIS FERNANDO YÉPEZ FUEL

DIRECTOR: Dr. RICHARD PAUL RIVERA GUEVARA

DMQ, febrero 2022

CERTIFICACIONES

Yo, ALEXIS FERNANDO YÉPEZ FUEL 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.



ALEXIS FERNANDO YÉPEZ FUEL

alexis.yeppez01@epn.edu.ec

alexixyepex@gmail.com

Certifico que el presente trabajo de integración curricular fue desarrollado por ALEXIS FERNANDO YÉPEZ FUEL, bajo mi supervisión.



PhD. RICHARD PAUL RIVERA GUEVARA

DIRECTOR

richard.rivera01@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante 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.

ALEXIS FERNANDO YÉPEZ FUEL

DEDICATORIA

Quiero dedicar este proyecto a aquellas personas que han formado parte de mi vida y han cumplido con un papel importante dentro de ella. En primer lugar, dedico este trabajo a mi Dios, por ser quien me ha permitido llegar hasta este momento de mi vida y poner en mi camino a personas con las que he podido contar cien por ciento a lo largo de esta experiencia. La más importante entre ellas, por supuesto, mi madre, quien ha sido la persona que ha estado a mi lado en todo momento y circunstancia, a Ismael, mi hermano, quien ha sido mi ejemplo a seguir de un hombre esforzado y valiente.

A Stephanie, mi hermana, mi mejor amiga y mi mayor tesoro en esta vida, por su puesto también se lo dedico a mi padre. Igualmente, dedico este trabajo a mi tía materna, Rosa, quien se ha encargado de brindarme su apoyo directa e indirectamente para poder culminar con esta carrera, así como también a una persona, que, a pesar de ya no encontrarse con vida en este mundo, tengo la seguridad que al día hoy estaría muy orgulloso en saber que he concluido con esta etapa de mi vida, pastor José Valencia.

AGRADECIMIENTO

La gratitud es un sentimiento que demuestra el valor que se le da a las acciones que los demás hacen en beneficio de nosotros, por esta razón, quiero expresar mis agradecimientos a las personas que me han acompañado durante el transcurso de mi carrera y me han brindado su apoyo para que la realización de este proyecto y lo que este representa se haya cumplido con éxito. Por supuesto, agradezco en primer lugar a Dios por darme la fortaleza, sabiduría, y sobre todo la paciencia para poder avanzar un paso a la vez en mi vida y mi carrera, sin su compañía y ayuda nada de esto hubiera sido posible.

A mi madre, por darme su mano en los momentos de mayor dificultad, y siempre estar pendiente de lo que pasa conmigo, su disciplina y su consuelo han sido factores importantes que me han ayudado a formarme como persona y como profesional. A mi hermano, Ismael, quien gracias a su ejemplo me ha enseñado lo que un verdadero hombre debe hacer en su hogar, además de ser el primer profesional de la familia y así dar el primer paso para que, como sus hermanos, hayamos decidido seguir el mismo camino. A mi hermana, Stephanie, quien en todo momento ha confiado en mí y me ha llenado de esperanza y esfuerzo con sus palabras de aliento. A mi padre, que a pesar de las circunstancias me ha hecho saber a su manera, el apoyo que ha tenido para mí.

De igual manera quiero agradecer a los ingenieros Richard Rivera, Byron Loarte, Ivonne Maldonado, Juan Pablo Zaldumbide y Edwin Salvador por los conocimientos impartidos, su pedagogía, paciencia y profesionalismo han sido factores relevantes que han garantizado mi aprendizaje y mi formación como un profesional de la informática. También quiero agradecer a todas aquellas personas, estudiantes como yo, que encontré al comenzar este camino, entre ellas las dos más importantes, Wendy y Carlos, quienes me apoyaron siempre desde el primer día que los conocí y gracias a ellos la experiencia de estudiar esta carrera universitaria fue amena y satisfactoria.

Finalmente, quiero agradecer a la Escuela Politécnica Nacional (EPN) y a la Escuela de Formación de Tecnólogos (ESFOT) por abrirme sus puertas y acogerme como uno de sus estudiantes durante estos años. Mi mayor agradecimiento será el representar a esta prestigiosa institución y demostrar el nivel y calidad de personas y profesionales que día a día se forman en ella. Ha sido para mí un gran honor cumplir con este objetivo en la compañía de tan valiosas personas.

¡Muchas gracias!

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE TABLAS.....	VII
ÍNDICE DE FIGURAS.....	VIII
RESUMEN.....	IX
ABSTRACT.....	X
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.....	5
2.1 Metodología de Desarrollo.....	5
2.2 Diseño de interfaces.....	9
2.3 Diseño de la arquitectura.....	10
2.4 Herramientas de desarrollo.....	12
3 RESULTADOS.....	15
3.1 <i>Sprint</i> 0. Configuración del ambiente de desarrollo.....	15
3.2 <i>Sprint</i> 1. Módulo de presentación de la información.....	19
3.3 <i>Sprint</i> 2. Módulo de organización de la información.....	26
3.4 <i>Sprint</i> 3. Módulo de servicios adicionales para el usuario.....	32
3.5 <i>Sprint</i> 4. Pruebas y despliegue a producción.....	37
4 Conclusiones.....	44
5 Recomendaciones.....	45
6 Referencias bibliográficas.....	47
7 ANEXOS.....	51
ANEXO I. Certificado de originalidad.....	51
ANEXO II. Manual Técnico.....	53

Recopilación de requerimientos.....	53
Historias de Usuario	54
<i>Product Backlog</i>	60
<i>Sprint Backlog</i>	61
Prototipos de Interfaces	66
Pruebas de Aceptación.....	70
ANEXO III. Manual de Usuario	80
ANEXO IV. Manual de Instalación.....	81

ÍNDICE DE TABLAS

TABLA I. Roles del proyecto	7
TABLA II. Historia de usuario Nro. 1 - Perfil de usuario e inicio de sesión.....	8
TABLA III. Herramientas para el desarrollo del Sistema Web	12
TABLA IV. Prueba de Aceptación Nro. 1 - Inicio de sesión	39
TABLA V. Recopilación de requerimientos	53
TABLA VI. Historia de usuario Nro. 2 - Visualización de pulsaciones.....	54
TABLA VII. Historia de usuario Nro. 3 - Visualización de capturas de pantalla	54
TABLA VIII. Historia de usuario Nro. 4 - Visualización de pulsaciones en sitios web	54
TABLA IX. Historia de usuario Nro. 5 - Marcar y desmarcar un registro como favorito.....	55
TABLA X. Historia de usuario Nro. 6 - Visualizar listado de clientes	55
TABLA XI. Historia de usuario Nro. 7 - Organización de información por parámetros	55
TABLA XII. Historia de usuario Nro. 8 - Búsqueda de registros por parámetros.....	56
TABLA XIII. Historia de usuario Nro. 9 - Visualización de registros con parámetros	56
TABLA XIV. Historia de usuario Nro. 10 - Archivar y desarchivar información	56
TABLA XV. Historia de usuario Nro. 11 - Crear categorías	57
TABLA XVI. Historia de usuario Nro. 12 - Buscador de palabras	57
TABLA XVII. Historia de usuario Nro. 13 - Descarga de imágenes	57
TABLA XVIII. Historia de usuario Nro. 14 - Gráfico estadístico registros-clientes.....	58
TABLA XIX. Historia de usuario Nro. 15 - Gráfico estadístico de tipos de registros	58
TABLA XX. Historia de usuario Nro. 16 - Agregar registros a una categoría.....	58
TABLA XXI. Historia de usuario Nro. 17 - Administrar categorías	59
TABLA XXII. Historia de usuario Nro. 18 - Filtro de registros por tipo	59
TABLA XXIII. <i>Product Backlog</i>	60
TABLA XXIV. <i>Sprint Backlog</i>	61
TABLA XXV. Prueba de Aceptación Nro. 2 - Visualizar pulsaciones.....	70
TABLA XXVI. Prueba de Aceptación Nro. 3 - Visualizar capturas de pantalla.....	70
TABLA XXVII. Prueba de Aceptación Nro. 4 - Visualizar pulsaciones de sitios web	71
TABLA XXVIII. Prueba de Aceptación Nro. 5 - Marcar y desmarcar un registro como favorito.....	71
TABLA XXIX. Prueba de Aceptación Nro. 6 - Visualizar listado de clientes	72
TABLA XXX. Prueba de Aceptación Nro. 7 - Ordenamiento de registros por parámetros	72
TABLA XXXI. Prueba de Aceptación Nro. 8 - Búsqueda de registros por parámetros.....	73
TABLA XXXII. Prueba de Aceptación Nro. 9 - Visualizar registros con sus parámetros..	73
TABLA XXXIII. Prueba de Aceptación Nro. 10 - Marcar y desmarcar un registro como archivado.....	74
TABLA XXXIV. Prueba de Aceptación Nro. 11 - Crear categorías	75
TABLA XXXV. Prueba de Aceptación Nro. 12 - Búsqueda de palabras	75
TABLA XXXVI. Prueba de Aceptación Nro. 13 - Descargar imágenes.....	76
TABLA XXXVII. Prueba de Aceptación Nro. 14 - Visualizar número de registros por cliente.....	76
TABLA XXXVIII. Prueba de Aceptación Nro. 15 - Visualizar número de registros por cliente y tipo.....	77
TABLA XXXIX. Prueba de Aceptación Nro. 16 - Agregar registro a una categoría.....	77
TABLA XL. Prueba de Aceptación Nro. 17 - Gestionar categorías.....	78
TABLA XLI. Prueba de Aceptación Nro. 18 - Filtrado de registros según su tipo	78

ÍNDICE DE FIGURAS

Fig. 1 Prototipo de interfaz - Página de inicio	10
Fig. 2 Modelo Vista Controlador	11
Fig. 3 Patrón arquitectónico	11
Fig. 4 Instalación de XAMPP	16
Fig. 5 Diseño de la base de datos	17
Fig. 6 Comando de instalación de Lavarel	17
Fig. 7 Comando de creación de proyecto en Laravel	17
Fig. 8 Estructura inicial del proyecto Laravel	18
Fig. 9 Comando para crear el proyecto de Next	18
Fig. 10 Estructura inicial del proyecto de Next	18
Fig. 11 Formulario de inicio de sesión	19
Fig. 12 Validación de campos del formulario	20
Fig. 13 Validación de credenciales	20
Fig. 14 Tabla de registros	21
Fig. 15 Columna de registro desplegada	22
Fig. 16 Registro de tipo captura de pantalla	23
Fig. 17 Registro de tipo sitio web	24
Fig. 18 Búsqueda por nombre de ventana	24
Fig. 19 Filtrado de registros por tipo captura de pantalla	25
Fig. 20 Tabla de visualización de clientes	26
Fig. 21 Información adicional de cliente desplegada	27
Fig. 22 Orden alfabético ascendente de registros	28
Fig. 23 Formulario de creación de categoría	29
Fig. 24 Lista de categorías	29
Fig. 25 Verificación de asignación a categoría	30
Fig. 26 Selector de categorías	31
Fig. 27 Gráfico estadístico registros-clientes	32
Fig. 28 Registros marcados como favoritos	33
Fig. 29 Tabla de registros archivados	34
Fig. 30 Descarga de captura de pantalla	35
Fig. 31 Gráficos estadísticos tipo de registro por cliente	35
Fig. 32 Validación de longitud de palabra	36
Fig. 33 Búsqueda sin resultados	36
Fig. 34 Resultados de búsqueda exitosa	37
Fig. 35 API Rest desplegada en Heroku	40
Fig. 36 Variables de entorno del API Rest	41
Fig. 37 Ejecución de migraciones en la base de datos	41
Fig. 38 Conexión de proyecto <i>frontend</i> con Vercel	42
Fig. 39 Variable de entorno del proyecto <i>frontend</i>	42
Fig. 40 Prototipo de interfaz - Formulario de inicio de sesión	66
Fig. 41 Prototipo de interfaz - Página de registros principal	66
Fig. 42 Prototipo de interfaz - Despliegue de registro de pulsaciones de teclas	67
Fig. 43 Prototipo de interfaz - Despliegue de registro de captura de pantalla	67
Fig. 44 Prototipo de interfaz - Página de registros archivados	68
Fig. 45 Prototipo de interfaz - Página de registros favoritos	68
Fig. 46 Prototipo de interfaz - Página de clientes	69

RESUMEN

El proyecto corresponde al desarrollo de un *dashboard* web que tiene el propósito de facilitar la visualización de los datos recabados por el *keylogger* creado para Windows 11. Este componente puede ser utilizado por un solo tipo de usuario, el administrador, cuenta con credenciales (correo electrónico y contraseña) únicas creadas para su acceso. No existe la opción de registrar nuevos usuarios dentro del sistema, por lo cual, se han almacenado previamente dichas credenciales en la base de datos.

El *dashboard* web propone solucionar el problema encontrado en cuanto al manejo de datos, si bien estos son obtenidos con éxito, el almacenarlos en un lugar específico y su presentación, son los dos principales inconvenientes que pretende solventar esta aplicación. Ha sido desarrollada como una aplicación distribuida que cuenta con tres principales capas, se definen de forma general como: interfaz de usuario, lógica del negocio y almacenamiento de datos. Cada una de estas capas, junto con los componentes que las conforman, han sido construidas por medio de herramientas, lenguajes de programación y marcos de trabajo dedicados como Next.js, Laravel y PostgreSQL respectivamente.

En este documento se detallan aspectos importantes involucrados en el desarrollo como: la implementación de la metodología ágil Scrum, la arquitectura de datos de la aplicación, la justificación de las herramientas usadas, los resultados obtenidos de la codificación de cada componente y la evaluación respectiva de funcionamiento de la aplicación en general. Por su puesto, se presentan conclusiones y recomendaciones pertinentes obtenidas a partir de la finalización de este proyecto.

PALABRAS CLAVE: desarrollo web, aplicación distribuida, código abierto, *backend*, *frontend*.

ABSTRACT

The project corresponds to the development of a web dashboard that has the purpose of facilitating the visualization of the data collected by the keylogger created for Windows 11. This component can be used only by one type of user, the administrator, with unique credentials (email and password) created for access. There is no option to register new users within the system, therefore these credentials have been previously stored in the database.

The web dashboard proposes to solve the problem encountered in terms of data management, although these are obtained successfully, storing them in a specific place and their presentation are the two main drawbacks that this application aims to solve. It has been developed as a distributed application with three main layers, generally defined as user interface, business logic, and data storage. Each one of the layers, together with the components that comprise them, have been built using dedicated tools, programming languages, and frameworks such as Next.js, Laravel, and PostgreSQL, respectively.

This document details important aspects involved in the development such as the implementation of the agile Scrum methodology, the data architecture of the application, the justification of the tools used, the results obtained from the coding of each component, and the respective evaluation of the app performance. Of course, conclusions and pertinent recommendations obtained from the completion of this project, are presented.

KEYWORDS: *web development, distributed application, open source, backend, frontend.*

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

El componente se ha desarrollado para que el usuario pueda interactuar con la información captada por el *keylogger* [1]. Esta interacción comprende principalmente la visualización de los registros, ya que el volumen de estos es considerable, se ha encontrado la necesidad de recopilar toda esta información en un solo lugar, una base de datos. El dashboard web se construye en base a tres funciones principales; almacenamiento, recepción, y presentación de los datos. Como se mencionó, los registros generados por el *keylogger* son almacenados en una base de datos relacional por medio de una API Rest, esta ha sido desarrollada con PHP, el cual es un lenguaje de programación de uso general comúnmente adaptado al desarrollo de sitios web [2].

El API tiene un papel de suma importancia en el funcionamiento del dashboard, ya que se ubica entre el almacenamiento de datos y la presentación de estos hacia el usuario final. Conforme a esto, es la encargada de ingresar los datos provenientes del *keylogger* en la base de datos, este flujo dentro del sistema es conocido como el *backend* de la aplicación ya que implica funcionalidades y características de ésta que para el usuario siempre están ocultas, a pesar de su interacción con la misma.

En cuanto a la visualización de los registros, el *dashboard* web cuenta con una interfaz gráfica para que el usuario pueda acceder a estos datos de manera sencilla, siempre y cuando, cuente con las credenciales establecidas para autenticarse en él. Esta parte de la aplicación ha sido desarrollada con JavaScript, un lenguaje de programación interpretado que, de forma nativa, funciona en todos los navegadores [3]. Por medio de esta interfaz gráfica se garantiza que el usuario pueda obtener un listado de los registros obtenidos del computador en el cual se haya alojado el *keylogger* previamente.

Todos los registros que se presentan tienen características propias y comunes, esto se ha definido de tal manera que el usuario pueda diferenciar entre ellos, de acuerdo con el contenido de cada uno. Los tres tipos de contenido que se manejan son: un registro con una captura de pantalla, otro con las teclas pulsadas y/o texto captado y otro que muestre las teclas pulsadas durante la visita a un sitio web. El número de registros captados y presentados de cada tipo son proporcionales a la interacción del usuario en el ordenador que tiene el *keylogger* instalado.

Además de esto, se han añadido acciones que el usuario puede ejecutar dentro de la aplicación, entre estas se tiene: la funcionalidad de búsqueda de palabras, descarga de capturas de pantalla, marcar o desmarcar un registro como favorito, añadir un registro a una categoría de favoritos, archivar/desarchivar un registro, gráficos estadísticos que

reflejan información rápida y en tiempo real, añadir categorías y ciertas opciones adicionales que se explican a detalle en una sección posterior de este documento.

1.1 Objetivo general

Desarrollar un dashboard web para el *keylogger* de Windows 11.

1.2 Objetivos específicos

1. Determinar los requerimientos funcionales y no funcionales para el dashboard web.
2. Diseñar el modelo de base de datos y la arquitectura del dashboard en base a los requerimientos obtenidos.
3. Codificar los componentes del sistema, *backend* y *frontend*, de acuerdo con los requerimientos obtenidos.
4. Comprobar el funcionamiento del dashboard web para su despliegue a producción.

1.3 Alcance

Para que este componente cubra las necesidades que implica la administración de los datos provistos por el *keylogger* [4], se ha establecido un análisis previo para definir el conjunto de herramientas a utilizar en el proceso de desarrollo del *dashboard* web. Estableciendo así, el uso del marco de trabajo Next.js, para la construcción del *frontend* de la aplicación. Este marco de trabajo hace uso de la librería de código abierto React, para brindar una experiencia completa de desarrollo a quien la utilice.

Además, se suman dependencias y librerías utilizadas para maquetar cada componente de la interfaz, por ejemplo, se ha utilizado Material-UI, una librería que brinda diferentes tipos de componentes acompañados de funcionalidades para hacer uso de ellos en React [5], Angular, Vue y demás librerías de JavaScript. Para el diseño y estilización de la parte visual, se ha implementado Styled Components, una dependencia utilizada para implementar diseños sobre componentes de manera casi nativa y similar a como se lo hace con CSS [6].

Para construir el API Rest se ha elegido Laravel como marco de trabajo. Laravel hace que la programación en PHP sea elegante y simple, además tiene una estructura de directorios bien definida logrando así que cada módulo y servicio que se cree dentro de él sea accedido eficientemente [7]. Para el contexto de información que se maneja dentro del *dashboard* web, es importante controlar que las transacciones de datos se realicen de manera rápida y en tiempo real, para lograr esto, se dispone de Eloquent.

Eloquent es el Object Relational Mapping (ORM) o Mapeo de Objeto Relacional que posee Laravel, este permite manejar el procesamiento en la base de datos de forma sencilla y provee funciones para realizar consultas y peticiones de datos complejas sin tener la necesidad de escribir código SQL para esto [8].

Los datos se administran por medio del Sistema Gestor de Base de Datos (SGBD) PostgreSQL. Se ha concluido en el uso de este ya que soporta un gran volumen de datos y es multiplataforma, además cuenta con los tipos de datos más utilizados actualmente [9]. El uso de esta tecnología en el desarrollo del *backend* se consolida en que esta forma parte de un modelo de software libre, ideal para la ejecución de este proyecto.

1.4 Marco Teórico

Un *keylogger* es un tipo de software malicioso [10], que pretende obtener información de los dispositivos que están infectados sin el consentimiento del usuario [11]. Actualmente, este tipo de malware puede infectar no solo computadoras, si no también dispositivos móviles [12], [13], puesto que en los últimos años se han desarrollado múltiples familias de este tipo de malware [14]. Además, los *keyloggers* pueden ser utilizados por los ciberdelincuentes para realizar espionaje, incluyéndolo con software malicioso más avanzado como las amenazas persistentes [15].

Para desarrollar software de calidad, uno de los primeros y más importantes aspectos que se deben considerar es el establecer una metodología sobre la cual se va a llevar a cabo el desarrollo. Se define a la metodología de desarrollo de software como un conjunto de técnicas y métodos aplicables para diseñar soluciones de software [16]. El objetivo principal que se busca al aplicar una metodología es conseguir organización en el equipo de trabajo [17]. Es algo irresponsable empezar el desarrollo de un proyecto de software sin una metodología, las consecuencias más comunes de hacer esto son los retrasos en tiempo de entrega, errores del sistema y problemas con la conclusión de procesos [17].

Las metodologías comúnmente usadas, hasta hace algunos años atrás, en el desarrollo de software eran las llamadas tradicionales. Una metodología tradicional se caracteriza por la definición total y rígida de requisitos al inicio de un proyecto. Tiene una estructura lineal, por lo cual, cada una de las etapas definidas en ella se deben completar una a continuación de la otra, es decir que, no se puede volver a una etapa después de que esta haya finalizado ni comenzar con la siguiente si la anterior no se ha concluido [17]. Entre las principales metodologías tradicionales están: cascada, prototipado, espiral, incremental y el diseño rápido de aplicaciones.

En la actualidad, las características de una metodología tradicional no son convenientes para el desarrollo de la mayoría de los proyectos de software, ya que los cambios son constantes y se requiere flexibilidad continuamente. Para satisfacer esta necesidad existen las metodologías ágiles, se denomina ágil a una metodología porque produce eficiencia y productividad en un equipo de trabajo, ya que este conoce en todo momento que es lo que tiene que hacer [17]. Si bien este tipo de metodologías están basadas en la metodología incremental, han sido adaptadas para optimizar los procesos en la construcción del producto final, este a su vez se obtiene en base a pequeñas funcionalidades que se agregan en cada ciclo [17].

Los ciclos tienen una corta duración, por lo cual, la existencia de cambios imprevistos es manejable y permite que el proyecto se adapte correcta y continuamente. Además, el cliente del software está involucrado directamente con el desarrollo del proyecto en tiempo real, permitiéndole así verificar el avance de este. Las iteraciones continuas son las que van “puliendo” el producto hasta conseguir un software funcional que cumpla con los requerimientos del cliente. Las metodologías ágiles más usadas hoy en día son: Scrum, Kanban, Lean y Programación extrema (XP) [17].

La estructura de un sistema web se puede establecer de manera general en dos aspectos principales, la interacción del usuario y el funcionamiento que este tiene por detrás. La interacción hace referencia a la presentación de los datos y el funcionamiento no visual incluye el almacenamiento, transformación, envío y recepción de estos. En palabras técnicas, estas dos partes se denominan como *frontend* y *backend* respectivamente.

El *frontend* es la capa de visualización de un sistema, esta le permite interactuar al usuario por medio de eventos, gráficos y componentes visuales útiles para brindarle una mejor experiencia en el manejo de datos [18]. Como ejemplo de esto se puede considerar un formulario de inicio de sesión en una red social, para que el usuario pueda acceder al contenido de esta red es necesario que se autentique, el *frontend* hace que la única acción que el usuario deba realizar para conseguir esto sea el ingresar un correo electrónico o nombre de usuario junto con una contraseña y pulsar un botón. Esto para el usuario es simple y no lo compromete a tener un conocimiento basto de tecnología más que el de recordar su contraseña y su nombre de usuario o correo electrónico.

Por otra parte, el *backend* es el encargado de la gestión y almacenamiento de los datos que se manejan dentro de un sistema. Es el componente que controla las transacciones, recibe solicitudes desde la interfaz y envía respuestas para su presentación [18]. Siguiendo

con el ejemplo anterior, el proceso de autenticación que está por detrás del ingreso de credenciales y pulsación de un botón es más complejo.

En este contexto, el flujo con el que el *backend* tiene que cumplir es el recibir la petición enviada desde la interfaz, validar las credenciales recibidas, verificar su concordancia con lo almacenado en la base de datos, recibir el resultado de la interacción con esta y enviar la respuesta correspondiente de vuelta hacia el cliente. Por lo tanto, si el usuario está registrado en la red social y sus credenciales son correctas está facultado para acceder al contenido de la red, si no es así, el contenido es restringido.

Para el desarrollo *backend* de un sistema actualmente existen herramientas tales como Java, Python, MySQL, PHP, C++, SQL Server, MongoDB, Laravel, entre otras. Por otro lado, para desarrollar *frontend* se puede hacer uso de herramientas como JavaScript, Angular, HTML, Backbone, React, CSS, TypeScript, entre las más comunes [18].

2 METODOLOGÍA

Este proyecto es un estudio de caso [19] basado en una prueba de concepto. Una prueba de concepto permite verificar la factibilidad de implantar una idea, método, aplicación o programa y validar el potencial que podría llegar a tener. En el caso del software conviene saber que el producto puede integrarse correctamente con los demás procesos para evaluar su uso, buscar mejoras y brindar detalles de los beneficios generados para el cliente, ayudándole así con la toma de decisiones sobre la aprobación de una idea y la implementación de ésta a mayor escala [20].

2.1 Metodología de Desarrollo

Para cumplir con el desarrollo de este proyecto, se ha propuesto un eje de construcción basado en la metodología ágil Scrum. Esta metodología consiste en un proceso general que acompañado de buenas prácticas, permite a un equipo de trabajo conseguir los objetivos trazados dentro del tiempo estimado [21]. Además, la experiencia de uso de esta metodología ha permitido comprobar que es útil para usarla en proyectos de desarrollo de software ya que la mayoría de estos presentan características en común, tales como: requisitos cambiantes o que no están bien definidos, los resultados que se esperan deben ser presentados a tiempo y llevar una flexibilidad obligatoria dentro de su construcción.

Scrum ha sido útil para dividir al proyecto en una serie de tareas alcanzables en períodos de tiempo relativamente cortos, a estos se los conoce como Sprints. El *Sprint* es un intervalo de tiempo que generalmente tiene la duración de una semana, hasta máximo un mes, esto dependiendo del tipo de proyecto en el que se lo esté definiendo. En cada *Sprint*

se establece una serie de actividades que se deben llevar a cabo para obtener al final de este, un entregable que forma parte del producto final [22]. Este proceso continuo garantiza que con el inicio y final de cada *Sprint* el producto incrementa en su desarrollo hasta llegar a su culminación [22]. Por lo citado, se ha implementado esta metodología en la construcción del *dashboard* web, los resultados obtenidos se detallan en una sección siguiente de este documento.

2.1.1 Roles

Product Owner

En Scrum, el *Product Owner* comprueba que el proyecto avance y es el encargado de verificar que el sistema que se está construyendo cumpla con los requerimientos planteados y este a su vez cuenta con las funcionalidades definidas en cada historia de usuario [23]. El Dr. Richard Rivera ha sido la persona encargada de cumplir con este rol durante el desarrollo del componente, ya que fue quien, en primera instancia, compartió la idea con el equipo de desarrollo y ha sido consciente del funcionamiento esperado con el que debe cumplir el componente para su aprobación.

Scrum Máster

Es el encargado de dar un seguimiento al equipo de desarrollo en el proyecto, tiene la tarea de verificar la implementación de teorías y reglas pertinentes que maneja Scrum [24]. En este caso, el rol de Scrum Máster también lo cumple el Dr. Richard Rivera debido a que es la persona quien ha estado al tanto de lo que pasa con el equipo de desarrollo, teniendo así que encargarse de solventar inconvenientes presentados durante el desarrollo del proyecto.

Equipo de desarrollo

El equipo de desarrollo para este proyecto está conformado por una persona, Alexis Yépez, quien como estudiante y programador se ha comprometido a llevar a cabo el conjunto de tareas designadas para cada Sprint, esto en busca de obtener un producto tangible con la finalización de cada etapa de trabajo y así haber llevado a cabo el desarrollo de este proyecto. La **TABLA I** presenta los roles establecidos dentro del desarrollo del proyecto siguiendo la metodología Scrum.

TABLA I. Roles del proyecto

ROL	INTEGRANTE
<i>Product Owner</i>	Dr. Richard Rivera
Scrum Máster	Dr. Richard Rivera
Equipo de desarrollo	Sr. Alexis Yépez

2.1.2 Artefactos

Los artefactos de Scrum son las herramientas físicas resultantes de la aplicación de esta metodología sobre un proyecto [25].

Recopilación de Requerimientos

Antes de comenzar a desarrollar un proyecto de software, es necesario levantar y conocer los requerimientos que este necesita, es una tarea fundamental que se debe cumplir por parte de todos los involucrados. Los requerimientos del *dashboard* web se han obtenido después de haber realizado el respectivo análisis junto con el *Product Owner* y el equipo de desarrollo. Estos se presentan con más detalle en el ANEXO II sección *Recopilación de requerimientos*.

Historias de Usuario

Una historia de usuario sirve para dar claridad al desarrollador sobre una funcionalidad que se quiera añadir a un sistema de software. Indica quién es el responsable de llevar a cabo esa tarea y que es lo que se espera obtener al finalizar la misma [26]. A continuación, la **TABLA II** es la representación de lo que es una Historia de usuario. El resto de estas se encuentran en este documento, en el ANEXO II sección *Historias de Usuario*.

TABLA II. Historia de usuario Nro. 1 - Perfil de usuario e inicio de sesión

HISTORIA DE USUARIO	
Identificador (ID): HU01	Usuario: Administrador del Dashboard
Nombre Historia: Perfil de usuario e inicio de sesión	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web tendrá un perfil de usuario con el que podrá iniciar sesión para acceder a este y a las funcionalidades que tiene.	
Observación: Este perfil de usuario será único y de tipo de administrador, por lo cual tendrá todos los permisos y acciones disponibles dentro del <i>dashboard</i> web.	

Product Backlog

El *Product Backlog* es un documento que recopila la lista de requisitos transformados en historias de usuario, por esta razón, es común que este artefacto presente constantes cambios a lo largo del proyecto ya que sobre él influyen varios factores, uno de los más significativos es el de los requisitos cambiantes o mal definidos. Es una herramienta de tal importancia, que es manejada por el *Product Owner*, convirtiéndola así en un eje principal del desarrollo de un proyecto [25]. El *Product Backlog* del *dashboard* web se encuentra en el ANEXO II sección *Product Backlog*.

Sprint Backlog

Para que el programador pueda cumplir con el objetivo planteado en una historia de usuario, esta debe ser desestructurada en tareas más pequeñas y específicas que en conjunto lo ayuden a alcanzar la nueva funcionalidad o característica del sistema. Es en este punto donde el *Sprint Backlog* pasa a formar parte de los artefactos de desarrollo, sirve para establecer cada una de las tareas que se deben llevar a cabo para, en primer lugar, obtener el resultado establecido, y en segundo, cumplir con el tiempo de duración propuesto para cada Sprint. De esta manera se define de manera explícita, que se debe hacer y cuando se lo debe hacer [25]. El *Sprint Backlog* definido para la construcción del *dashboard* web se lo puede visualizar en el ANEXO II sección *Sprint Backlog*.

2.2 Diseño de interfaces

Concluida la etapa de levantamiento de requisitos del sistema web, es conveniente proceder con el diseño de este. A continuación, se describe la herramienta utilizada para este fin junto con un ejemplo del resultado obtenido.

2.2.1 Figma

Para diseñar los *mockups* del *dashboard* web, se ha utilizado la herramienta Figma. Esta es una herramienta dedicada para el diseño y prototipado de interfaces que cuenta con funcionalidades que la hacen, no solo cumplir con la creación completa de una interfaz de usuario, sino que también se puede establecer en ella el flujo esperado para la aplicación [27]. De esta manera Figma facilita el manejo de *User Interface* (UI) [28] y *User Xperience* (UX) [29] en un proyecto de software.

Está basada en el navegador principalmente, lo que hace que esté disponible en todo momento para su uso. Contiene un contexto de trabajo en equipo completo y brinda bibliotecas de herramientas útiles, así como también la posibilidad de crear nuevas [27]. Por lo mencionado, se ha hecho uso de Figma para la construcción de los prototipos del *dashboard*.

A continuación, la **Fig. 1** presenta un ejemplo de los prototipos construidos para el *dashboard* web. El resto de estos se encuentran dentro del ANEXO II sección *Prototipos de Interfaces*.

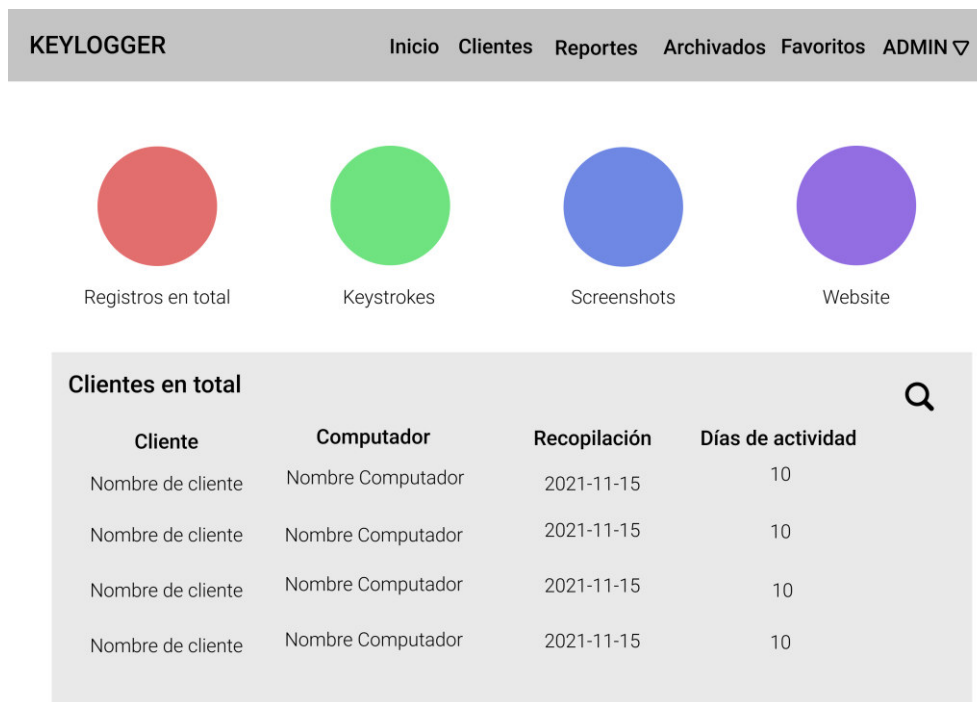


Fig. 1 Prototipo de interfaz - Página de inicio

2.3 Diseño de la arquitectura

La elección del patrón para establecer la arquitectura de los datos es importante en un proyecto de software, esta depende especialmente del tipo de sistema que se quiere construir. En cuanto al desarrollo del *dashboard* web, se ha propuesto definir al patrón arquitectónico Modelo Vista Controlador (MVC) como base de la arquitectura de datos.

2.3.1 Patrón arquitectónico

El patrón arquitectónico MVC es un tipo de arquitectura de software que permite separar a un sistema en tres componentes para que el manejo de datos sea simple. Los componentes que lo conforman son: la interfaz de usuario, la lógica de control y el modelo de datos [30].

El modelo se encarga de acceder a la capa de almacenamiento y definir las reglas del negocio para la información, según el tipo de modelo, este puede acceder a la vista directamente. Comúnmente se define que el modelo tenga relación únicamente con su controlador para el flujo de almacenamiento [30], esta es la forma que se ha establecido en el *dashboard* web.

Como parte del patrón MVC, **el controlador** es el que se encuentra entre los datos y la interfaz de usuario. Se encarga de gestionar el flujo de información entre ambos, logra este propósito ya que cuenta con funcionalidades y reglas definidas que le permiten adaptarse

durante la comunicación con cada parte [30]. Un ejemplo de esto es una petición realizada desde la interfaz de usuario por medio de un evento, si suponemos que el evento es una petición de datos específica, el controlador es el encargado de recibir esa petición, evaluar que el modelo de datos la contenga, comunicarse con él y esperar su respuesta para después procesarla y enviársela de vuelta a la interfaz de usuario, mejor conocida como vista.

Finalmente, **la vista** o interfaz de usuario es el componente del MVC que interactúa directamente con el usuario y le presenta a este la información provista por los dos componentes ya mencionados. Mediante la interfaz es que el usuario final puede ejecutar acciones sobre el sistema, al hacerlo sucede un proceso de envío y transacción de datos y peticiones totalmente transparente para él [30]. La **Fig. 2** muestra la estructura del patrón arquitectónico MVC.



Fig. 2 Modelo Vista Controlador

A continuación, se presenta la **Fig. 3**, misma que representa el patrón arquitectónico MVC aplicado en la construcción del *dashboard web*.

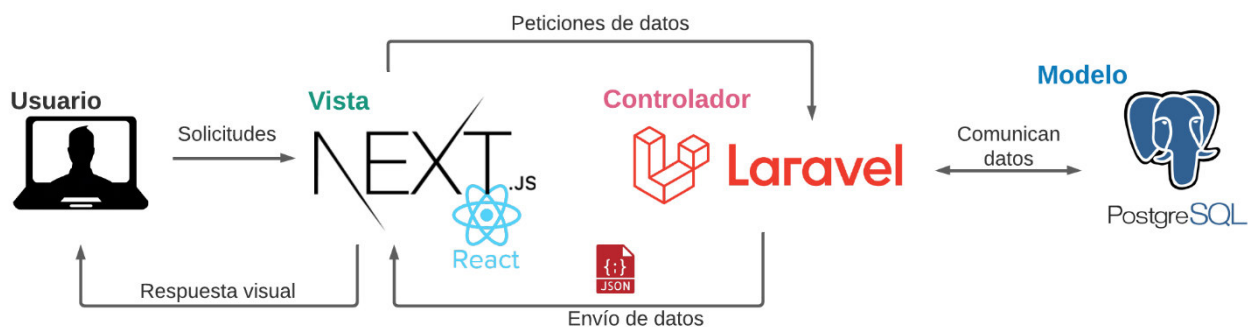


Fig. 3 Patrón arquitectónico

2.4 Herramientas de desarrollo

La **TABLA III** presenta en su contenido las herramientas, descripción y el porqué de su utilización en el desarrollo del *dashboard* web.

TABLA III. Herramientas para el desarrollo del Sistema Web

Herramienta	Descripción	Justificación
PostgreSQL	Es un sistema gestor de base de datos de código abierto que permite trabajar con bases de datos relacionales. Soporta consultas en lenguaje SQL para el modelo relacional y de tipo JSON para consultas no relacionales [9].	Se ha decidido utilizar PostgreSQL como gestor de almacenamiento ya que este soporta un gran volumen de datos. También por ser multiplataforma y contar con tipos de datos más utilizados actualmente. Además, es un SGBD de código abierto, ideal para el funcionamiento del proyecto.
XAMPP	XAMPP es una herramienta provista por Apache, que a su vez contiene herramientas como MySQL/MariaDB, PHP, Apache, y Perl. Es una aplicación completa que facilita el desarrollo de sistemas web [31].	Se ha utilizado XAMPP por la facilidad que ofrece al momento de instalar servicios necesarios para el desarrollo, principalmente brinda una ventaja ante el proceso de instalación tradicional de cada uno de los servicios mencionados.
Postman	Postman es una aplicación que sirve para cumplir con pruebas de APIs Rest, cuenta con otras funciones, pero su uso más común es el de testear, consumir y depurar sistemas <i>backend</i> [32].	Para garantizar el funcionamiento del <i>backend</i> de la aplicación web, es necesario realizar pruebas conforme las funcionalidades se van creando, por lo cual se ha utilizado Postman para simular las peticiones de un cliente y su interacción con los datos.
Laravel	Laravel es un marco de trabajo de PHP que sirve para construir aplicaciones web, cuenta con los recursos necesarios para crear una aplicación completa gracias a sus funciones predefinidas y sistema de paquetes [7].	Se ha hecho uso de Laravel en el desarrollo del API por la facilidad que ofrece al implementar el patrón arquitectónico MVC.
PhpStorm	Es un <i>Integrated Development Enviroment</i> (IDE) compatible con PHP, brinda la función de autocompletado y depuración de código en tiempo real. Soporta la integración con HTML, CSS y JavaScript [33].	Si bien se ha podido utilizar cualquier editor de código para la programación del API, se ha elegido PhpStorm por ser un IDE dedicado y optimizado para este propósito. Cuenta con funcionalidades que facilitan la codificación junto con paquetes adicionales que se pueden instalar para mejorar la experiencia de desarrollo.
NodeJS	Es el entorno de ejecución, en tiempo real, de JavaScript. Este incluye todos los recursos necesarios para ejecutar programas	El uso de NodeJS facilita la instalación de librerías de JavaScript como React, además soporta el funcionamiento de Next.js. Esto hace

	desarrollados con este lenguaje de programación [34].	que el desarrollo del <i>frontend</i> sea más eficiente.
Next.js	Es un marco de trabajo que permite a los desarrolladores crear aplicaciones de alto rendimiento, su principal atributo para alcanzar esto es que usa la renderización en el lado del servidor [35].	Next.js es útil para desarrollar el <i>frontend</i> del <i>dashboard</i> ya que este presenta datos en tiempo real, por lo cual se requiere que este proceso de consulta y presentación de datos se ejecute rápidamente.
React	React es una librería de JavaScript utilizada para crear interfaces de usuario. Cuenta con un entorno completo de herramientas, módulos y componentes que potencian aún más el desarrollo <i>frontend</i> [36].	El marco de trabajo Next.js hace uso de React para su funcionamiento, por lo cual se lo ha utilizado en la creación de componentes de las pantallas del <i>dashboard</i> web.
Material-UI	Material-UI es una de las principales librerías de interfaz de usuario usadas en React, provee de componentes preconstruídos con funciones embebidas que facilitan el desarrollo de una aplicación, tanto para el programador como para el usuario [5].	Para facilitar la maquetación de cada interfaz dentro del <i>dashboard</i> , se ha hecho uso de esta librería ya que provee componentes ya construidos muy completos. Una ventaja es que a estos se los puede estilizar a disposición.
MUI-Datatables	Es un componente creado por Material-UI, se lo considera como una herramienta ya que cuenta con funcionalidades que ningún tipo de tabla provista por otras librerías tiene.	Se ha elegido este componente dentro de la aplicación para la presentación principal de los datos, esto porque es un componente personalizable y viene con funciones predefinidas útiles y optimizadas para el manejo de una gran cantidad de datos.
Styled Components	Se trata de una librería que sirve para dar estilos CSS a componentes de React. Entre sus características principales están: mejora del mantenimiento, evitar sobrecarga de clases, eliminación de CSS no utilizable [6].	Para diseñar la interfaz y estilizarla de mejor manera se ha usado esta librería. Su uso es simple y cercano al lenguaje CSS nativo, así como su mantenimiento ya que no es necesario definir nombres de clases para dar estilos a los componentes.
Chart JS	Biblioteca de JavaScript que sirve para presentar datos de manera gráfica por medio de figuras estadísticas. Cuenta con tipos de gráficos como: circular, barras, dona, pastel, lineal, entre otros [37].	El uso de esta biblioteca es sencillo e intuitivo, personalizable en ciertos aspectos y con funciones útiles y simples que facilitan el desarrollo. Eficaz para cumplir con la representación de tipos de datos que se desea en el <i>dashboard</i> .
WebStorm	WebStorm es un IDE perteneciente a la familia de JetBrains, especializado en la programación del lenguaje JavaScript. Es un editor inteligente lleno de funciones y atajos que mejoran significativamente la experiencia de desarrollo para el programador [38].	Se ha hecho uso de WebStorm como IDE para el desarrollo del <i>frontend</i> por la gran cantidad de herramientas que provee, el autocompletado y el análisis de código hace que se obtenga un código fuente más limpio y de calidad.

GitHub	GitHub es una plataforma de desarrollo de software mundialmente utilizada para el control de versiones en proyectos de software. Cuenta con múltiples opciones y configuraciones que hacen que el trabajo en equipo se simplifique y se mantenga controlado en todo momento [39].	Para manejar el avance del proyecto, tanto <i>frontend</i> como <i>backend</i> , se ha hecho uso de GitHub para mantener un flujo de trabajo limpio y fácil de seguir. Añadir una funcionalidad al proyecto principal resulta sencillo haciendo uso de esta herramienta.
---------------	---	--

3 RESULTADOS

Dentro de esta sección se presentarán los resultados obtenidos de cada Sprint, el desarrollo del *dashboard* junto con las funcionalidades esperadas se ha dividido en tres Sprints principales y el *Sprint 0* de configuración que se detalla a continuación.

3.1 *Sprint 0*. Configuración del ambiente de desarrollo

La definición de este *Sprint* ha sido necesaria para llevar a cabo las tareas correspondientes al acondicionamiento del ambiente de desarrollo. Esto ha sido importante para que la codificación del *dashboard* siga su flujo y este no sea interrumpido por falta de preparación antes de empezar. Los resultados obtenidos de este *Sprint* han sido las siguientes:

- Instalación de herramientas de desarrollo.
- Diseño y creación de la base de datos.
- Creación del proyecto en Laravel.
- Creación del proyecto en Next.

3.1.1 Instalación de herramientas de desarrollo

Para desarrollar el API y la interfaz gráfica del *dashboard* web se han instalado las siguientes herramientas:

XAMPP

Es útil la instalación de esta herramienta ya que cuenta con MySQL, PHP y Apache, servicios y paquetes, que, de manera tradicional, se tendrían que haber instalado uno por uno. Al descargar y ejecutar el instalador de XAMPP hay que seleccionar los componentes que se necesitan, la **Fig. 4** muestra la selección de los paquetes a instalar. Enfocados con el fin del proyecto, se eligen los tres ya mencionados.

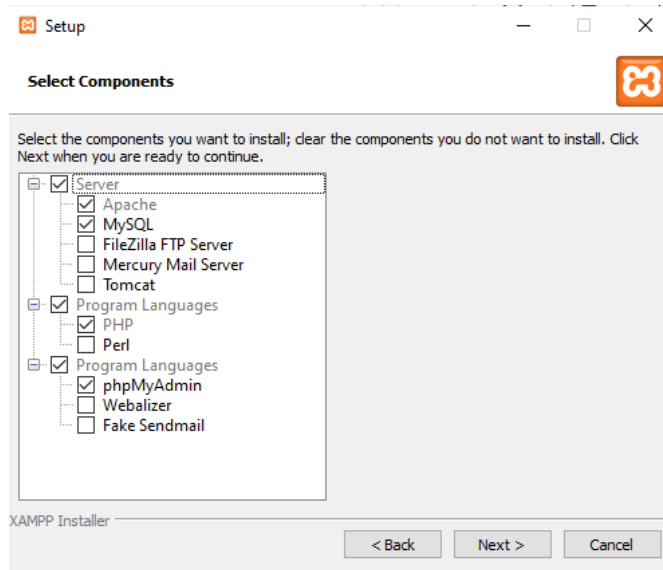


Fig. 4 Instalación de XAMPP

PhpStorm

El instalador de PhpStorm se encuentra en la página oficial de JetBrains, se ejecuta después de descargarlo y se siguen los pasos que muestra en pantalla. Se requiere de una licencia para hacer uso de este producto, por lo cual es necesario ingresar credenciales que cuenten con una, en este caso se ha validado el registro con una licencia de tipo estudiantil.

WebStorm

El proceso de instalación para WebStorm, el IDE usado para el *frontend*, es exactamente el mismo. Ya que también forma parte de los productos de JetBrains, se requiere de la licencia antes mencionada para su funcionamiento.

Postman

El instalador de Postman se lo puede descargar desde su página oficial, el proceso de instalación es el que se realiza por defecto.

3.1.2 Diseño y creación de la base de datos

La base de datos para el funcionamiento del dashboard web es de tipo relacional y ha sido definida para que tenga una tabla de usuarios, una de registros, otra de clientes y una final de categorías de favoritos. Este modelo sirve para el flujo de datos requerido en el *dashboard*. Su diseño se presenta a continuación en la **Fig. 5**.

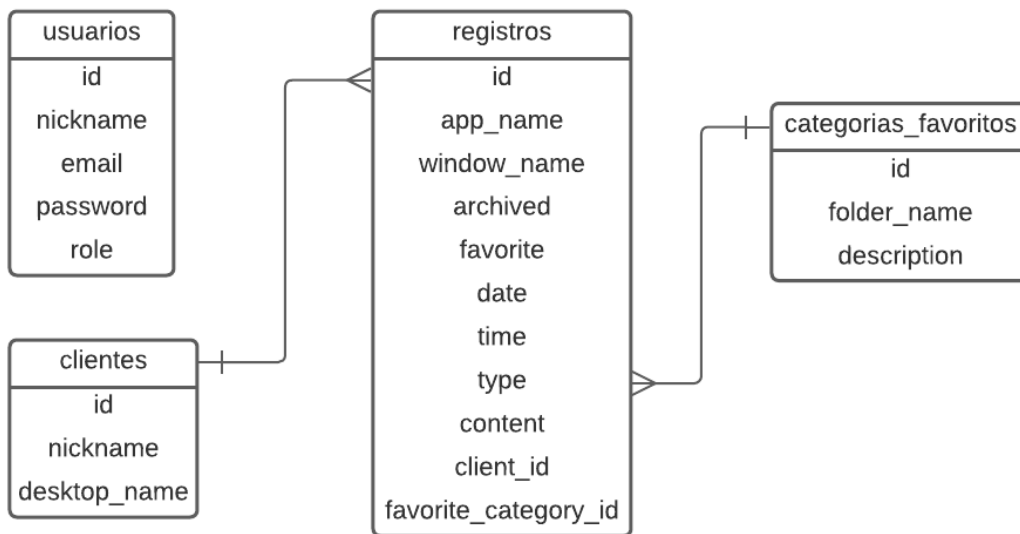


Fig. 5 Diseño de la base de datos

3.1.3 Creación del proyecto en Laravel

Para instalar Laravel en el sistema operativo Windows se utiliza el gestor de dependencias de PHP, Composer [40]. La **Fig. 6** presenta el comando a ejecutar para comenzar con la instalación de Laravel.

```
C:\Users\alexi>composer global require laravel/installer
```

Fig. 6 Comando de instalación de Laravel

Una vez instalado es posible crear el proyecto del API, la **Fig. 7** muestra el comando usado para esto.

```
C:\Users\alexi>laravel new api-keylogger
```

Fig. 7 Comando de creación de proyecto en Laravel

Una vez creado el nuevo proyecto de Laravel, este se inicia con el IDE PhpStorm. La **Fig. 8** muestra la estructura de directorios inicial del proyecto.

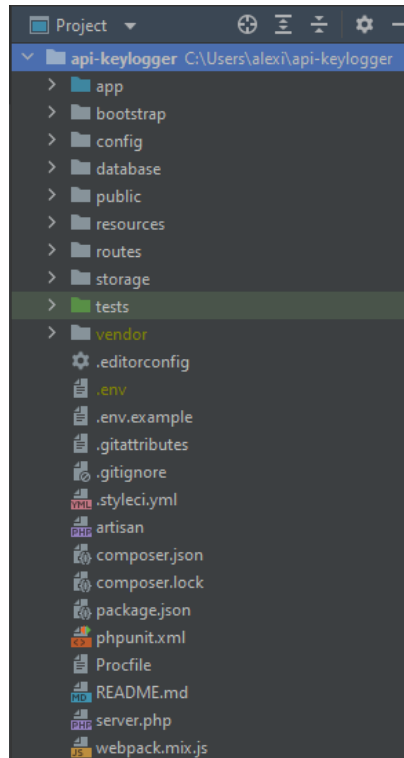


Fig. 8 Estructura inicial del proyecto Laravel

3.1.4 Creación del proyecto Next

Se crea el proyecto base para el *frontend* en Next ejecutando el comando que se muestra a continuación en la **Fig. 9**.

```
C:\Users\alexi>npx create-next-app Next.js-DashboardKeylogger
```

Fig. 9 Comando para crear el proyecto de Next

Una vez creado, se lo puede abrir con el IDE WebStorm. La **Fig. 10** presenta la estructura inicial del proyecto de Next.

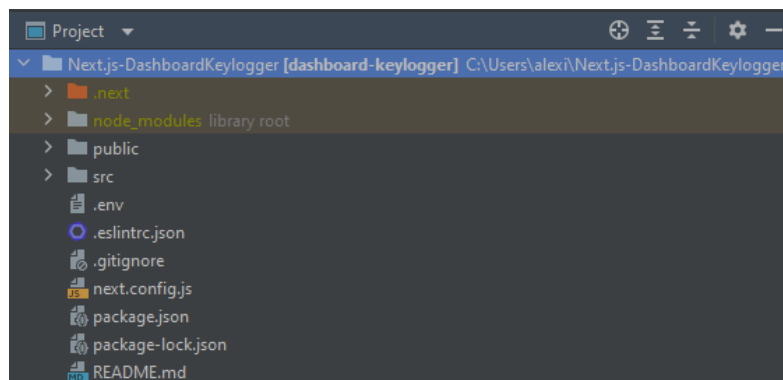


Fig. 10 Estructura inicial del proyecto de Next

3.2 *Sprint* 1. Módulo de presentación de la información

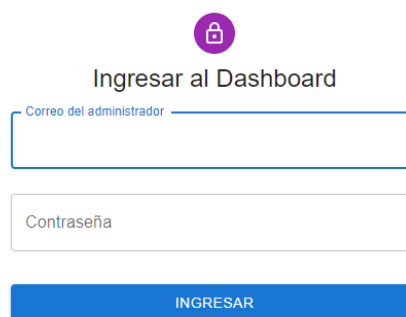
El *Sprint* 1 se ha establecido de manera que al final de este se pueda visualizar la información capturada por el *keylogger* en el *dashboard* web. A continuación, se listan los resultados obtenidos después de haber cumplido con las tareas designadas en este *Sprint*.

- Perfil de usuario y autenticación dentro del *dashboard*.
- Presentación de registros con parámetros específicos.
- Visualización de pulsaciones de teclas.
- Visualización de capturas de pantalla.
- Visualización de pulsaciones en sitios web visitados.
- Búsqueda de registros por parámetros.
- Filtro de registros según su tipo.

3.2.1 Perfil de usuario y autenticación dentro del dashboard

Formulario y validaciones

Para que el usuario administrador pueda acceder al *dashboard* se ha codificado un formulario de inicio de sesión. La **Fig. 11** muestra dicho formulario el cual cuenta con un campo para ingreso del correo, otro para la contraseña y un botón para ingresar.



El formulario de inicio de sesión está centrado y contiene lo siguiente:

- Un icono de candado morado en un círculo.
- El título "Ingresar al Dashboard" en un color gris oscuro.
- Un campo de entrada de texto con el placeholder "Correo del administrador" y un borde azul.
- Un campo de entrada de texto con el placeholder "Contraseña" y un borde gris.
- Un botón rectangular azul con el texto "INGRESAR" en blanco.

Keylogger 2022.

Fig. 11 Formulario de inicio de sesión

En el formulario de inicio de sesión se han implementado ciertas validaciones de los campos para garantizar la tolerancia a errores dentro del sistema. Se ha hecho uso de la

librería Yup [41] para validar el contenido en cuanto a campos obligatorios y tipos de datos ingresados. En caso de no pasar alguna de estas validaciones se le indica esto al usuario por medio de texto u otro componente, un ejemplo de esto lo presenta a continuación la **Fig. 12**.

Keylogger 2022.

Fig. 12 Validación de campos del formulario

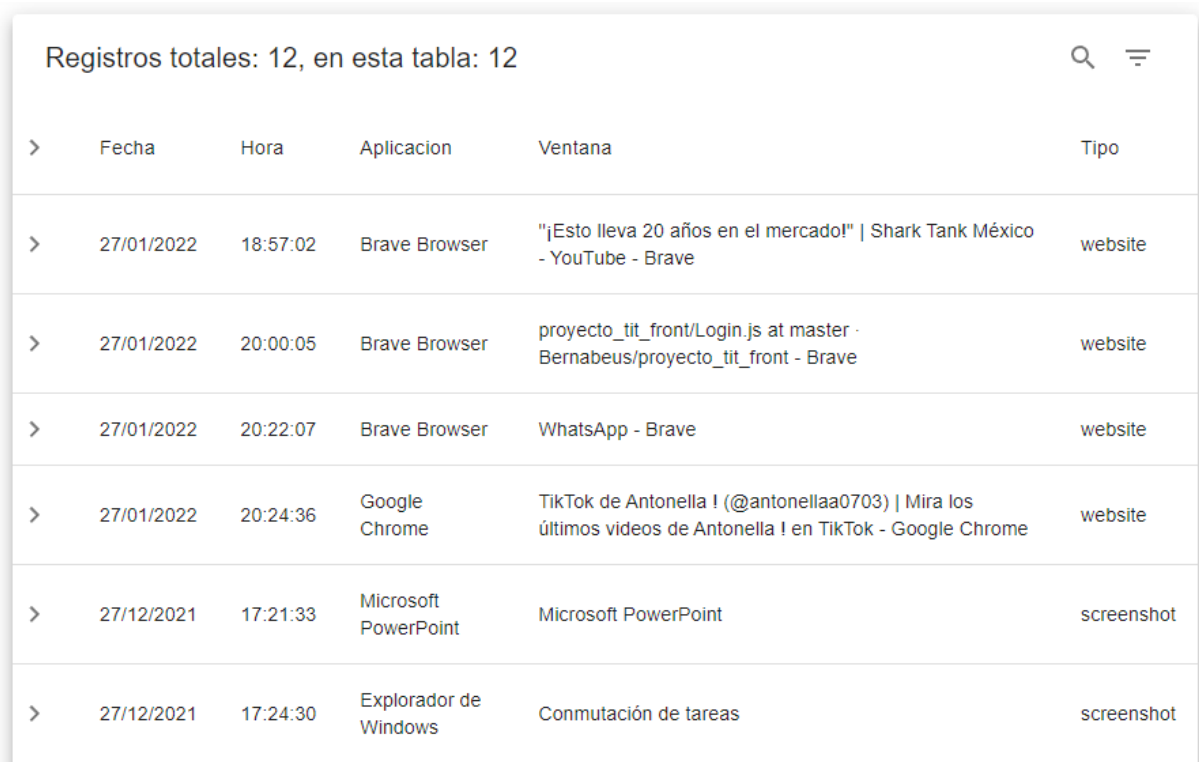
Otra validación de suma importancia es la de comprobar que las credenciales ingresadas sean congruentes con las almacenadas en la base de datos, para lograr esto se ha hecho uso de las funciones construidas en el API. Si el usuario ingresa correctamente el correo y la contraseña provistos se iniciará la sesión y accederá al dashboard, caso contrario, se mostrará un aviso como se muestra en la **Fig. 13**.

Keylogger 2022.

Fig. 13 Validación de credenciales

3.2.2 Presentación de registros con parámetros específicos

Para presentar la lista de registros se ha utilizado un componente de la librería Material UI, una tabla que incluye ciertas funcionalidades predefinidas que convenientemente se adaptan a las previstas en la planificación de este proyecto. Los atributos que acompañan a cada registro han sido escogidos de manera que brinden información útil al usuario acerca de cómo y cuándo ha sido obtenido cada uno. Los atributos en cuestión son: fecha, hora, nombre de aplicación, título de ventana y el tipo de registro. La **Fig. 14** muestra la presentación de registros dentro de la tabla.



Registros totales: 12, en esta tabla: 12					🔍	☰
>	Fecha	Hora	Aplicacion	Ventana	Tipo	
>	27/01/2022	18:57:02	Brave Browser	"¡Esto lleva 20 años en el mercado!" Shark Tank México - YouTube - Brave	website	
>	27/01/2022	20:00:05	Brave Browser	proyecto_tit_front/Login.js at master · Bernabeus/proyecto_tit_front - Brave	website	
>	27/01/2022	20:22:07	Brave Browser	WhatsApp - Brave	website	
>	27/01/2022	20:24:36	Google Chrome	TikTok de Antonella ! (@antonellaa0703) Mira los últimos videos de Antonella ! en TikTok - Google Chrome	website	
>	27/12/2021	17:21:33	Microsoft PowerPoint	Microsoft PowerPoint	screenshot	
>	27/12/2021	17:24:30	Explorador de Windows	Comutación de tareas	screenshot	

Fig. 14 Tabla de registros

Como se observa en la **Fig. 14**, cada registro cuenta con parámetros específicos, lo que representa cada uno se detalla a continuación:

- *Fecha*: Este campo indica la fecha en la que fue capturado el registro.
- *Hora*: Campo que indica la hora en la que fue capturado el registro.
- *Aplicación*: Este campo indica el nombre de la aplicación en la que fue capturado el registro.
- *Ventana*: Este campo contiene el título de ventana de la aplicación en la cual fue capturado el registro.

- *Tipo*: Indica de que tipo es el registro, este influye en el contenido que tendrá el mismo.

3.2.3 Visualización de pulsaciones de teclas

Si bien todos los registros tienen atributos generales, es importante saber que el contenido de cada uno difiere según su tipo. Para la visualización de los registros que contienen datos de las teclas pulsadas se ha usado un área de texto simple, está no se muestra en la primera forma de la tabla general, sino que se encuentra dentro de cada fila. Para acceder a ella basta con pulsar sobre cualquier área de la fila del registro, esto desplegará el resto de información contenida. A continuación, la **Fig. 15** muestra lo mencionado.

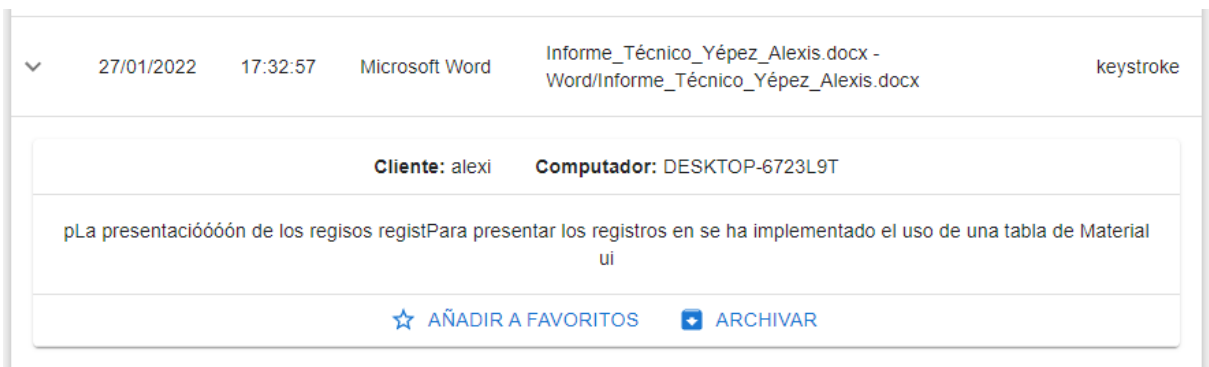


Fig. 15 Columna de registro desplegada

Esta funcionalidad es provista por la tabla y facilita la visualización general de los registros, ya que si el contenido de cada registro varía en longitud, este no afecta en la primera presentación de la tabla porque su despliegue es un evento controlado por el usuario. La información adicional con la que cuenta cada registro es el nombre de cliente, es decir, el nombre de usuario que haya iniciado sesión en el computador donde se encuentra instalado el *keylogger* [42]. También se muestra el nombre del computador y un par de botones con funciones que se explican en una sección posterior de este documento.

3.2.4 Visualización de capturas de pantalla

El manejo de imágenes dentro de la tabla se implementa de manera similar, se hace uso del componente *Image* [43] que provee *Next*. Esto hace que la renderización de cada captura de pantalla sea rápida ya que este componente cuenta con una optimización para la presentación de imágenes, es importante evaluar el tipo de registro antes de enviarlo a la tabla ya que el componente *Image* solo puede ser completado con un recurso que contenga una imagen, por ejemplo, una url, un código en base64 o una ruta de acceso

hacia el almacenamiento en donde se encuentre esta. Un ejemplo de registro de captura de pantalla se muestra a continuación en la **Fig. 16**.

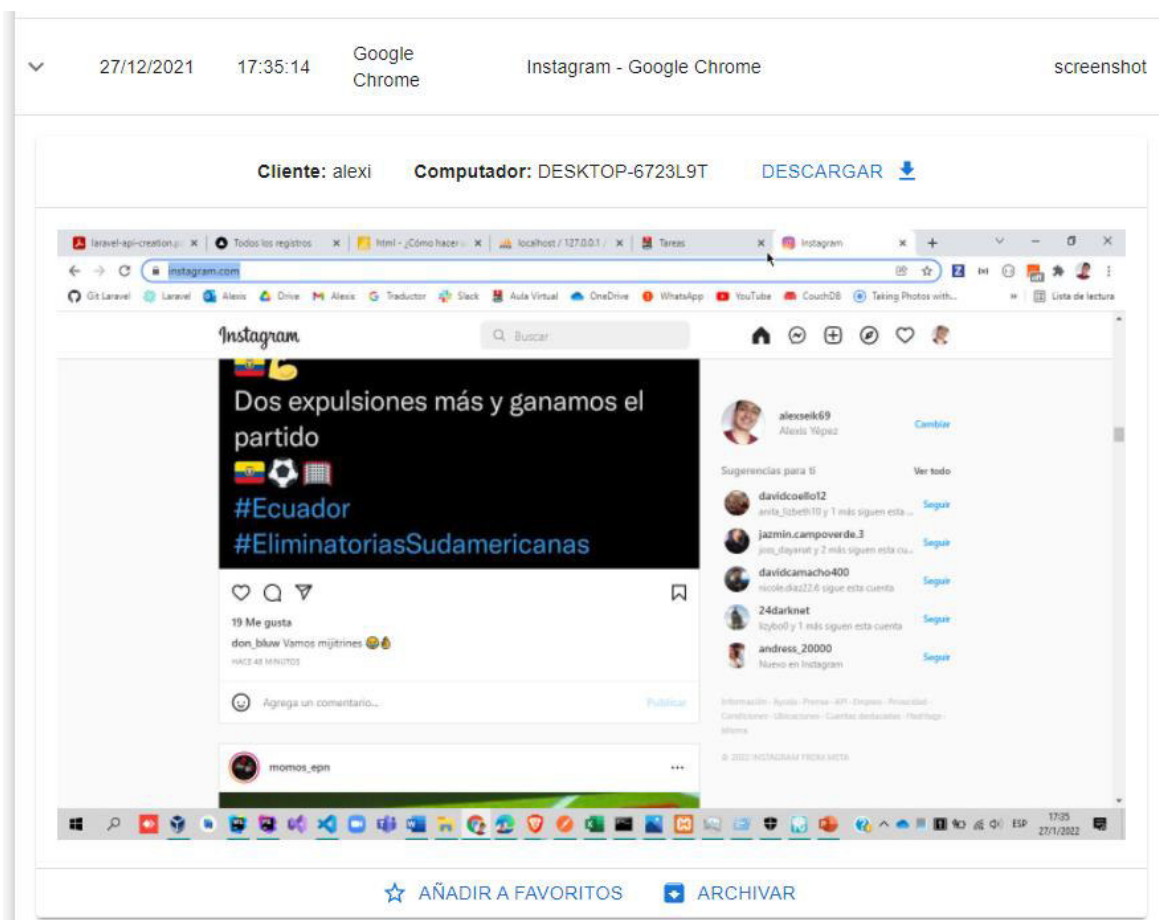


Fig. 16 Registro de tipo captura de pantalla

La **Fig. 16** muestra la presentación de un registro de tipo captura de pantalla para el usuario, aquí es donde se observa con claridad la ventaja que brinda el despliegue del contenido controlado, ya que la captura tiene dimensiones considerables y no sería conveniente presentarla en la primera vista de la tabla general. También se percibe que la celda cuenta con un tercer botón adicional a los vistos en el registro de pulsaciones de teclas, esto se debe a que cada registro de tipo captura de pantalla cuenta con la opción de descargar su contenido, es decir, la imagen. Esta funcionalidad se detalla en una sección más adelante.

3.2.5 Visualización de pulsaciones en sitios web visitados

La presentación de los registros de tipo sitio web es implementada bajo las mismas validaciones dentro de la tabla, es decir que, al desplegar un registro de este tipo se encontrarán las pulsaciones de teclas como contenido. Se ha definido este tipo de registro

para obtener diferenciación entre los registros de pulsaciones creados en un navegador y los obtenidos en el resto de las aplicaciones usadas por el cliente. La **Fig. 17** representa un registro de tipo sitio web.

—	Fecha	Hora	Aplicacion	Ventana	Tipo
▼	27/01/2022	12:40:32	Google Chrome	Tareas - Google Chrome	website
<p style="text-align: center;">Cliente: joel Computador: DESKTOP-86624MJD</p> <p style="text-align: center;">organizador de tareas online</p> <p style="text-align: center;"> ★ AÑADIR A FAVORITOS 📁 ARCHIVAR </p>					

Fig. 17 Registro de tipo sitio web

Un detalle importante con el que cuenta este tipo de registro es que en la columna “Ventana” se almacena el título del sitio web acompañado del nombre del navegador en el que se haya visitado dicho sitio. Esto implica que la mayoría de los registros que se capturen en un navegador serán clasificados como de tipo sitio web frente a los otros dos tipos existentes.

3.2.6 Búsqueda de registros por parámetros

El resultado obtenido de esta tarea ha sido la funcionalidad de búsqueda en los registros según sus parámetros, esta función se ha conseguido gracias a una de las tantas configuraciones con las que cuenta esta tabla. Para esto se ha definido con exactitud la estructura y el contenido de cada columna, ya que la búsqueda se basa en lo que cada celda (intersección de una columna con una fila en la vista principal) tiene. Los parámetros por los que se puede realizar una búsqueda son: nombre de aplicación, título de ventana, fecha y hora. La **Fig. 18** presenta un ejemplo de lo mencionado.

<input type="text" value="whatsapp"/> 🔍 ☰					
>	Fecha	Hora	Aplicacion	Ventana	Tipo
>	27/01/2022	20:22:07	Brave Browser	WhatsApp - Brave	website
Jump to Page: 1 ▾ Rows per page: 10 ▾ 1-1 of 1 < >					

Fig. 18 Búsqueda por nombre de ventana

Las **Fig. 18** muestra el resultado de realizar una búsqueda de la palabra “whatsapp”, la función ha coincidido este término con el título de ventana de un registro. Como se observa, la búsqueda es generalizada y no difiere entre mayúsculas o minúsculas. También es posible realizar la búsqueda ingresando números ya que los datos presentados en la tabla han sido transformados en una cadena de caracteres, por lo cual, la función evalúa la existencia de la cadena ingresada y devuelve todas las coincidencias.

3.2.7 Filtro de registros según su tipo

El filtrado de registros es una función que tiene la tabla general y ha sido definida de tal manera que presente todos los registros que sean del tipo que se requiera. El usuario puede acceder a este filtrado de datos por medio del botón que se encuentra en la esquina superior derecha, al presionarlo se desplegará un pequeño menú con las opciones de filtración.

Entre estas opciones se encuentran los tres tipos de registros que se maneja en el *dashboard*; pulsaciones de teclas, capturas de pantalla y sitios web. Esto resulta útil cuando el usuario necesita visualizar registros de un tipo específico, la tabla presenta solo los registros que coincidan con el tipo que se haya elegido. La **Fig. 19** representa un ejemplo de esto:

The image shows a table with a filter 'screenshot' applied. The table has columns for Fecha, Hora, Aplicacion, Ventana, and Tipo. There are four rows of data, all with 'screenshot' in the Tipo column. The bottom of the table shows pagination controls: 'Jump to Page: 1', 'Rows per page: 10', and '1-4 of 4'.

>	Fecha	Hora	Aplicacion	Ventana	Tipo
>	27/12/2021	17:21:33	Microsoft PowerPoint	Microsoft PowerPoint	screenshot
>	27/12/2021	17:24:30	Explorador de Windows	Conmutación de tareas	screenshot
>	27/12/2021	17:35:14	Google Chrome	Instagram - Google Chrome	screenshot
>	27/12/2021	19:25:14	Microsoft Teams	General (TSDS - Proyecto de Titulación 2021B) Microsoft Teams	screenshot

Jump to Page: 1 Rows per page: 10 1-4 of 4 < >

Fig. 19 Filtrado de registros por tipo captura de pantalla

3.3 *Sprint* 2. Módulo de organización de la información

Con la visualización de registros completada y siguiendo la planificación establecida en el *Sprint Backlog*, en este segundo *Sprint* se han definido tareas enfocadas a la visualización y gestión del resto de información provista por el API. Los resultados obtenidos al finalizar con este *Sprint* se resumen en:

- Visualización de clientes
- Organización de registros por parámetros específicos
- Creación de categorías
- Agregar registro a una categoría
- Gestión de categorías
- Gráfico estadístico relación registros-clientes

3.3.1 Visualización de clientes

Como se mencionó en un punto anterior, el término “cliente” dentro del contexto del *dashboard* hace referencia al nombre de usuario que tiene una sesión activa en el computador donde el *keylogger* se encuentra instalado. Esta información es primordial ya que permite identificar a los registros que se almacenan en la base de datos y a su vez le da conocimiento al usuario por medio de la interfaz web. Para conseguir esto se ha utilizado el mismo componente tabla que fue presentado en la sección Presentación de registros con parámetros específicos, sin embargo, la estructura y el contenido de datos que se presentan es diferente, esto lo muestra a continuación la **Fig. 20**.



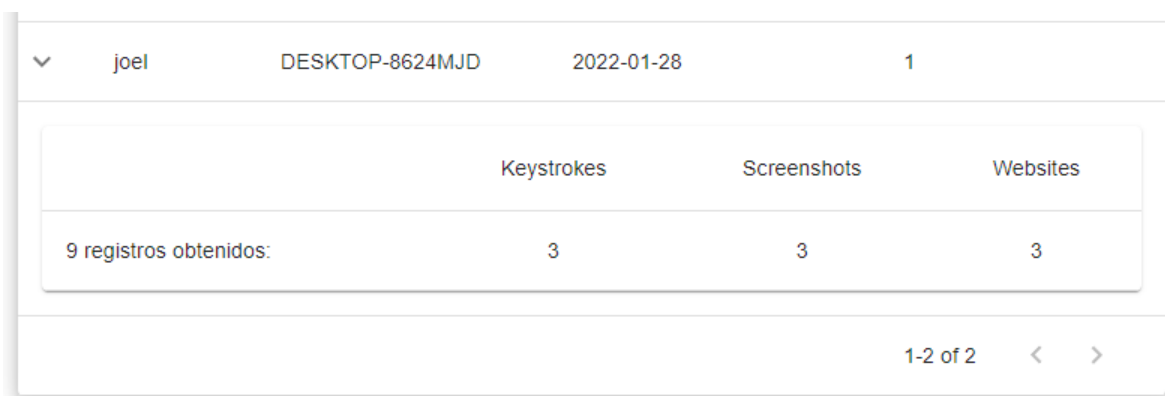
>	Cliente	Computador	Inicio de recopilación	Días de actividad
>	alexi	DESKTOP-6723L9T	2022-01-27	2
>	joel	DESKTOP-8624MJD	2022-01-28	1

Fig. 20 Tabla de visualización de clientes

Lo que representa cada columna y su contenido se indica a continuación:

- *Cliente*: El nombre de usuario con sesión activa en el computador de donde se obtienen los registros.
- *Computador*: Nombre del computador en el que se ha instalado el *keylogger*.
- *Inicio de recopilación*: Este campo indica la fecha en la que el *keylogger* ha comenzado a capturar información de ese cliente.
- *Días de actividad*: El campo muestra el total de días que el *keylogger* lleva captando información desde su instalación hasta la fecha actual.

Esta tabla también implementa la función de despliegue de cada fila, el contenido del interior en este caso presenta el número total de registros que se han obtenido del cliente, así como el número de registros de cada tipo. Se observa la representación de lo mencionado en la **Fig. 21**.



The screenshot shows a table row with the following data: a dropdown arrow, the name 'joel', the computer name 'DESKTOP-8624MJD', the date '2022-01-28', and the number '1'. Below this row, a dropdown menu is open, displaying a table with the following structure:

	Keystrokes	Screenshots	Websites
9 registros obtenidos:	3	3	3

At the bottom right of the table, there is a pagination indicator '1-2 of 2' and navigation arrows.

Fig. 21 Información adicional de cliente desplegada

3.3.2 Organización de registros por parámetros específicos

En busca de una mejor experiencia del usuario, se ha implementado dentro de la tabla de visualización de registros la funcionalidad de ordenamiento. Esto ha requerido de una correcta definición de las cabeceras de la tabla, el usuario tiene la posibilidad de dar un nuevo orden, ascendente o descendente, a la presentación de registros de acuerdo con el parámetro que mejor le convenga. Los parámetros por los que el usuario puede cambiar el orden en la tabla son: la fecha, hora, nombre de aplicación y título de ventana. La **Fig. 22** muestra un ejemplo del orden alfabético aplicado a los registros según el campo "Ventana".

>	Fecha	Hora	Aplicacion	Ventana ↑	Tipo
>	27/01/2022	18:57:02	Brave Browser	"¡Esto lleva 20 años en el mercado!" Shark Tank México - YouTube - Brave	website
>	27/01/2022	17:25:56	Google Chrome	(7) Facebook - Google Chrome	keystroke
>	27/01/2022	17:24:30	Explorador de Windows	Conmutación de tareas	screenshot
>	28/01/2022	10:52:34	Microsoft Edge	Facebook - Trabajo: Microsoft? Edge	website
>	28/01/2022	10:40:08	Microsoft Edge	Facebook - Trabajo: Microsoft? Edge	keystroke
>	27/01/2022	19:25:14	Microsoft Teams	General (TSDS - Proyecto de Titulación 2021B) Microsoft Teams	screenshot
>	27/01/2022	17:32:57	Microsoft Word	Informe_Técnico_Yépez_Alexis.docx - Word/Informe_Técnico_Yépez_Alexis.docx	keystroke
>	27/01/2022	17:35:14	Google Chrome	Instagram - Google Chrome	screenshot
>	28/01/2022	12:40:16	Microsoft Excel	Libro1 - Excel	screenshot
>	28/01/2022	12:40:32	Microsoft Excel	Libro1 - Excel	keystroke

Jump to Page: 1 ▾ Rows per page: 10 ▾ 1-10 of 21 < >

Fig. 22 Orden alfabético ascendente de registros

3.3.3 Creación de categorías

Otra de las funcionalidades añadidas es la creación de categorías. Las categorías están enfocadas a que el usuario administrador del dashboard tenga la opción de clasificar de mejor manera aquellos registros que quiera marcar como favoritos, el cómo se puede marcar un registro como favorito se detalla en una sección posterior de este documento. Para crear una nueva categoría se ha requerido de un formulario que el usuario pueda completar para enviar esos datos como una petición al API. La **Fig. 23** muestra el formulario para crear una nueva categoría.

Nueva Categoría

Especifique un nombre y una descripción (opcional) para la nueva categoría

Nombre categoría

Redes sociales

Descripción

Aquí se almacenarán los registros que tengan relación con alguna red social

CREAR CANCELAR

Fig. 23 Formulario de creación de categoría

Como se observa, este formulario cuenta con dos campos, en el primero se ingresa el nombre que tendrá la nueva categoría y en el segundo se detalla una descripción de esta. En este formulario existen validaciones tanto del lado de la interfaz de usuario como en el API, entre estas se tiene que el título de una categoría no puede repetirse, es decir que, cada categoría es única dentro de la base de datos. También, el campo título no puede contener más de doscientos cincuenta y cinco caracteres ya que el tipo de dato definido no lo permite y principalmente es un campo obligatorio.

Por estas razones, se ha debido implementar validaciones a nivel de interfaz, por lo cual, si el usuario intenta enviar el formulario con el campo “Nombre de categoría” vacío, esto no tendrá ningún efecto y se le mostrará un mensaje de lo ocurrido. Esto no pasa cuando el usuario intenta crear la nueva categoría con el campo “Descripción” vacío ya que como se indica, este es opcional. En la **Fig. 24** se observa la lista de categorías de favoritos.

Seleccione una categoría

No asignar categoría

- Redes Sociales
- Sitios comunes
- Youtube

NUEVA CATEGORÍA CANCELAR AÑADIR

Fig. 24 Lista de categorías

Si los datos ingresados han pasado las validaciones pertinentes, se creará la nueva categoría y esta se reflejará en el listado de categorías que se presenta al usuario cuando desea marcar un registro como favorito.

3.3.4 Agregar registro a una categoría

Puesto que las categorías tienen el propósito de organizar de mejor manera los registros que el usuario considera más importantes, se ha incluido esta funcionalidad dentro del botón “Añadir a favoritos” que tiene cada registro. Al presionar dicho botón, se le presenta al usuario el listado con las categorías existentes, este tiene que elegir la categoría a la cual va a asignar el registro abierto. También cuenta con la posibilidad de crear una nueva categoría si así lo desea.

Al enviar la categoría seleccionada, el API se encarga de relacionar el registro con esta. Se puede verificar esta relación en el apartado de Favoritos del dashboard donde se encuentra el nuevo registro de la categoría. A continuación, la **Fig. 25** presenta el registro añadido a la categoría “Redes sociales”.

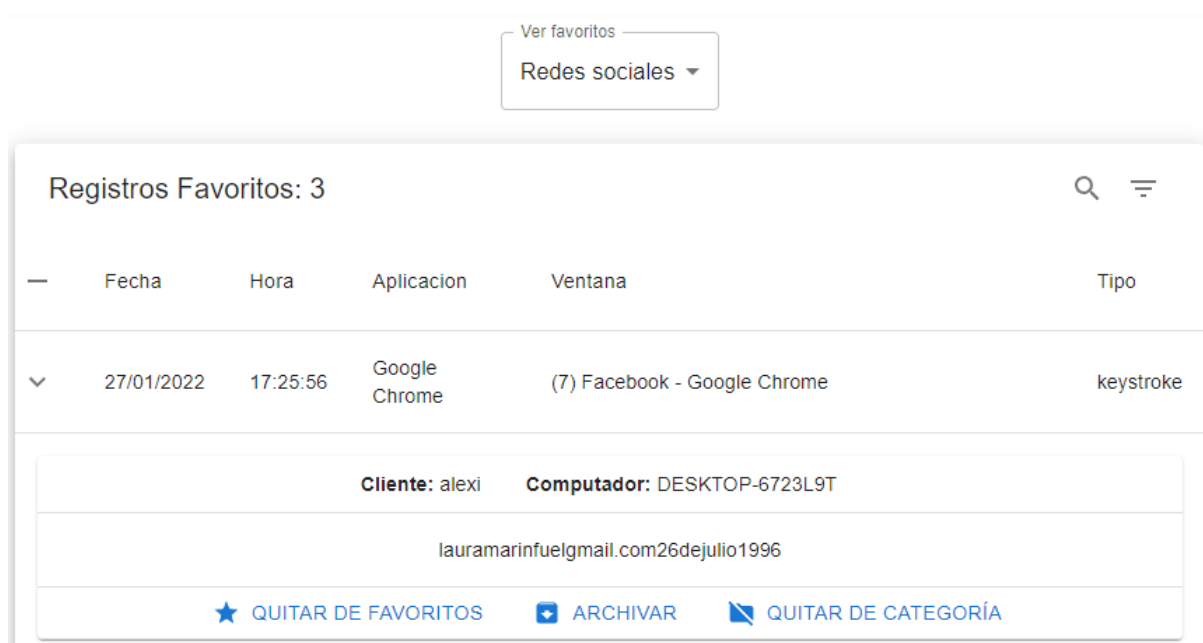


Fig. 25 Verificación de asignación a categoría

Como lo muestra la **Fig. 25**, el despliegue de la fila del registro refleja la existencia de tres botones con propósitos distintos, lo que hacen los dos primeros botones se explica más adelante en otra sección de este documento. El tercer botón es el que le permite al usuario eliminar al registro de la categoría en la que actualmente se encuentra, al presionarlo se envía una petición que indica que el proceso de asignación se revierta y la relación del

registro con la categoría se rompa. En consecuencia, este registro es considerado como favorito, pero sin una categoría asignada, por lo cual seguirá siendo accesible desde el apartado Favoritos.

3.3.5 Gestión de categorías

El usuario administrador puede acceder fácilmente a los registros pertenecientes a una categoría por medio de un componente selector que se ha implementado dentro del apartado de Favoritos. Este selector obtiene el nombre de las categorías por medio de una llamada al API, el flujo que se establece con este componente es que el usuario pueda seleccionar una categoría y en la tabla de registros favoritos se muestren todos aquellos que pertenezcan a esta. En la **Fig. 26** se muestra el despliegue del selector, este presenta cada categoría como una opción que puede elegir el usuario.

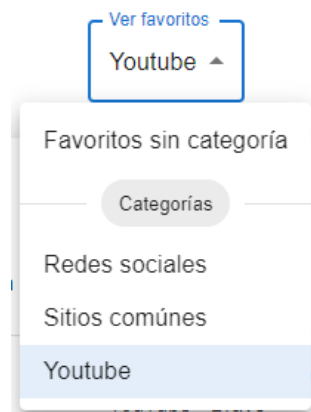


Fig. 26 Selector de categorías

3.3.6 Gráfico estadístico relación registros-clientes

Una manera útil de presentar los datos además de las tablas ya mencionadas es el manejar cierta información por medio de gráficos estadísticos. Para llevar a cabo esto se ha hecho uso de la librería Chart JS [37], una librería de gráficos estadísticos de uso simple y funcional. Uno de los cuatro gráficos con los que cuenta el *dashboard* es el que presenta información acerca del número de registros que se han obtenido por cada cliente. La **Fig. 27** muestra la presentación de este en la página inicial.

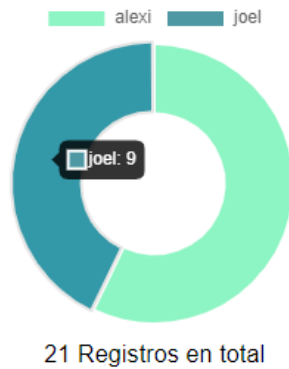


Fig. 27 Gráfico estadístico registros-clientes

Este gráfico indica el número de registros en total que ha captado el *keylogger* y la cantidad de estos por cada cliente, la información se actualiza cuando detecta un cambio en la base de datos y se refleja inmediatamente en la proporción del gráfico, el número de registros y la etiqueta, esto acompañado de una gama de colores alterna para diferenciar a los clientes con cada nueva actualización.

3.4 *Sprint* 3. Módulo de servicios adicionales para el usuario

Los resultados esperados para este *Sprint* han tenido el propósito de dar ciertas opciones adicionales al usuario en cuanto a lo que puede hacer con la información, de las tareas realizadas, se ha obtenido lo siguiente:

- Marcar un registro como favorito
- Archivar registros
- Descarga de imágenes
- Buscador de palabras
- Gráfico estadístico de tipos de registros

3.4.1 Marcar un registro como favorito

Es posible que se presente el caso en el que el usuario administrador desee marcar un registro como importante pero no necesite asignarlo a ninguna categoría específica para esto, para manejar este escenario se ha implementado la funcionalidad de marcar un registro como favorito. El funcionamiento de esto es similar al flujo que se lleva a cabo para añadir un registro a una categoría, por lo tanto, se puede acceder a esta opción por medio del botón “Añadir a favoritos”.

Esto desplegará la lista de categorías, sin embargo, la opción que debe ser escogida en este caso es la primera que no forma parte de la lista, “No asignar categoría”. Esto se puede observar en la **Fig. 24**. Esta acción le indica al API que debe marcar al registro escogido como favorito, pero sin que pertenezca a ninguna categoría. Para verificar que el proceso se ha completado, se accede al apartado de Favoritos del dashboard y aquí se presentarán por defecto aquellos registros que hayan sido marcados como favoritos sin categoría. La **Fig. 28** presenta lo mencionado.

Ver favoritos
Favoritos sin categoría ▾

Registros Favoritos: 3

—	Fecha	Hora	Aplicacion	Ventana	Tipo
▾	28/01/2022	12:40:32	Microsoft Excel	Libro1 - Excel	keystroke
<p>Ciente: joel Computador: DESKTOP-8624MJD</p> <p>actividades realizadas horan euvas tareas lunes enviar correo</p> <p>★ QUITAR DE FAVORITOS 📁 ARCHIVAR</p>					
>	28/01/2022	12:50:16	Postman	Postman	screenshot
>	28/01/2022	12:40:16	Microsoft Excel	Libro1 - Excel	screenshot

Fig. 28 Registros marcados como favoritos

El mismo selector se encarga de presentar este tipo de favoritos, se observa que el registro cuenta con dos botones, “Quitar de favoritos” y “Archivar”, este primero es el que le permite al usuario desmarcar a un registro como favorito. El presionarlo hace que se envíe al registro de vuelta a la lista de registros principal, esto por medio de una petición al API que se encarga de quitarlo de la lista de favoritos en la base de datos. Este botón es el mismo que se presenta en los registros que pertenecen a una categoría, por lo cual, el resultado de presionarlo en estos registros será el mismo. Esto se muestra en la **Fig. 25**.

3.4.2 Archivar registros

Así como el usuario puede diferenciar entre registros que le parezcan más importantes y enviarlos al apartado de Favoritos, también puede llegar a considerar que alguno de ellos contiene información irrelevante y no necesite tenerlo dentro de la lista principal. Para este

caso se ha añadido la opción de archivar un registro, esta funcionalidad se encuentra detrás del botón “Archivar” que tiene cada uno, al presionarlo, se ejecuta una petición al API encargada de marcar a dicho registro como archivado. El usuario puede acceder a este tipo de registros dentro del apartado de Archivados en el *dashboard*. La **Fig. 29** muestra la tabla de registros que han sido archivados.

Registros Archivados: 3						🔍	☰
—	Fecha	Hora	Aplicacion	Ventana		Tipo	
▼	28/01/2022	10:52:34	Microsoft Edge	Facebook - Trabajo: Microsoft? Edge		website	
<div style="border: 1px solid #ccc; padding: 5px; margin-bottom: 5px;"> <p>Cliente: joel Computador: DESKTOP-8624MJD</p> <p>https://www.facebook.com/photo/?fbid=119608840604899&set=gm.1370606580123251</p> <p>▶ DESARCHIVAR REGISTRO</p> </div>							
>	28/01/2022	12:25:59	Google Chrome	localhost / 127.0.0.1 phpMyAdmin 5.1.1 - Google Chrome		website	

Fig. 29 Tabla de registros archivados

Existe la opción de desarchivar un registro, esto hace que el apartado funcione de manera similar a como lo hace una papelera de reciclaje en un computador. Al presionar el botón “Desarchivar registro” se actualiza la tabla y ya no lo presenta, este registro vuelve al listado principal y allí puede ser localizado por el usuario nuevamente.

3.4.3 Descarga de imágenes

Como se detalló en la sección Visualización de capturas de pantalla, los registros de este tipo cuentan con la opción de descarga. Esto se ha implementado a nivel de interfaz por medio de una función que crea un objeto URL de acuerdo con el recurso de imagen que reciba, también recibe dos parámetros adicionales para definir el título de descarga de la imagen. Estos parámetros que recibe la función son el título de ventana y fecha del registro. A continuación, se presenta en la **Fig. 30** el resultado de presionar el botón de descarga de un registro.

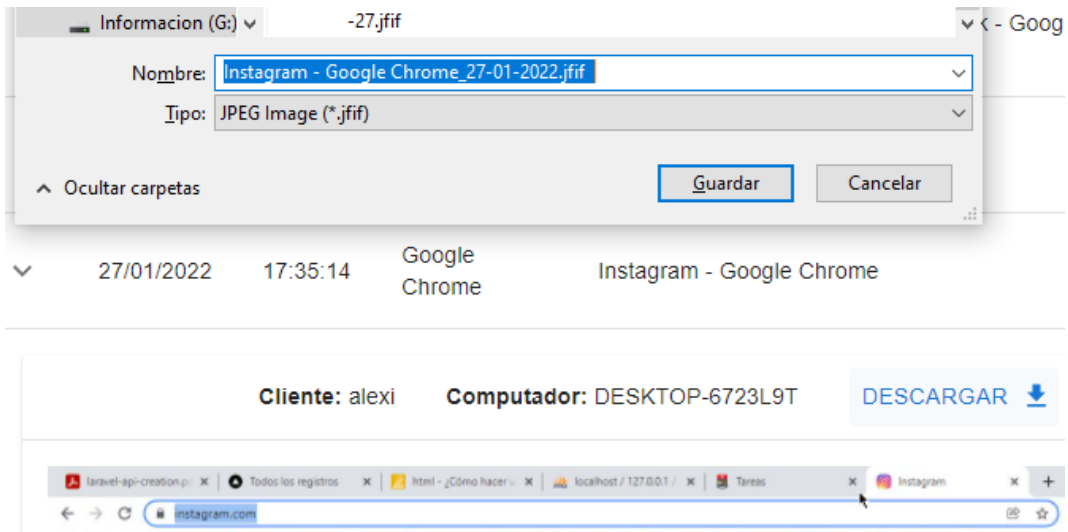


Fig. 30 Descarga de captura de pantalla

Se observa en la **Fig. 30** que el nombre por defecto de la captura a descargar es una concatenación de los dos parámetros mencionados, sin embargo, en el cuadro de dialogo del sistema el usuario puede definir el nombre que desee para continuar con la descarga.

3.4.4 Gráfico estadístico de tipos de registros

Otro de los gráficos utilizados en el *dashboard* es el de tipo pastel, provisto por la misma librería, cuenta con sencillez en el manejo, presentación y actualización de los datos. Se han implementado tres de estos gráficos correspondientes a los tres tipos de registros que se manejan. La **Fig. 31** indica el uso de estos gráficos.

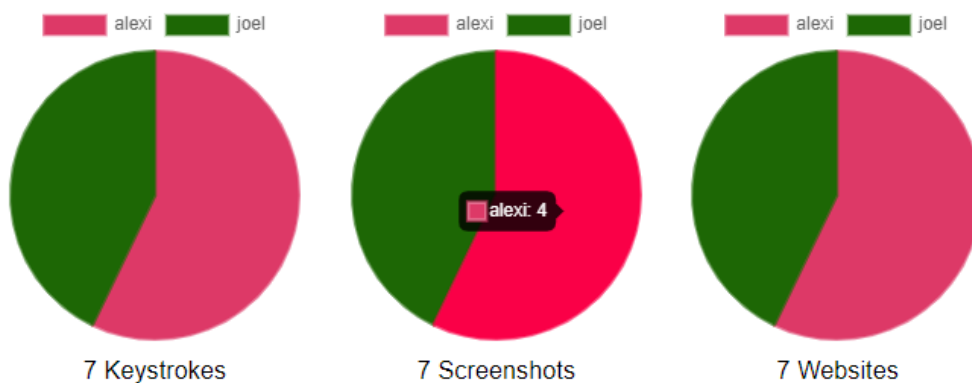


Fig. 31 Gráficos estadísticos tipo de registro por cliente

La etiqueta inferior muestra el número total de registros de ese tipo que actualmente se presentan en el *dashboard*, la proporción de cada uno indica la cantidad de registros de ese tipo que se han obtenido de cada cliente. En la **Fig. 31**, el gráfico de la mitad muestra

que siete de todos los registros son de tipo capturas de pantalla y cuatro de estos han sido capturados del cliente “alexi”.

3.4.5 Buscador de palabras

Otro de los resultados finales de este *Sprint* ha sido el obtener un buscador de palabras, esta funcionalidad le brinda al usuario la opción de ejecutar una búsqueda dentro del contenido de los registros de tipo pulsaciones de teclas. Se ha implementado esto ya que, como se explicó en la sección Búsqueda de registros por parámetros, el buscador propio de la tabla no incluye aquellos parámetros que se encuentren dentro de la fila del registro, es decir, la información que se presenta cuando este se despliega.

Este buscador ha sido implementado como un formulario para incluir validaciones dentro de él, una de ellas es que, si el usuario presiona el botón buscar sin haber ingresado ninguna palabra en el cuadro de texto, no se ejecute ningún evento. La **Fig. 32** muestra la validación que tiene el buscador en cuanto a la longitud de la palabra que se haya ingresado, esta debe ser de al menos cuatro caracteres para que se pueda ejecutar el proceso de búsqueda.



Fig. 32 Validación de longitud de palabra

Si el término cumple con esta validación se ejecuta la búsqueda, esta incluye una petición al API que retorna todos los registros que sean de tipo pulsaciones de teclas para su posterior comparación con la palabra ingresada. Si no se ha encontrado ninguna coincidencia dentro de los registros, se presenta un mensaje como lo muestra la **Fig. 33** a continuación.

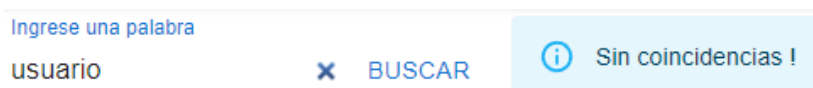


Fig. 33 Búsqueda sin resultados

Por otra parte, si existen palabras dentro del contenido de los registros que sean concordantes con el término de búsqueda, la tabla los presentará con su contenido desplegado para comprobar que verdaderamente se han obtenido resultados coincidentes. En la **Fig. 34** se observa que la palabra “redes” se encuentra en dos de los registros, para

volver al listado principal el usuario debe pulsar el botón “Reset” que se encuentra al lado derecho del campo de texto.

The screenshot shows a search interface with a search bar containing 'redes' and a 'RESET' button. A green notification box indicates '2 coincidencias encontradas'. Below the search bar, a table displays search results. The table has columns for Fecha, Hora, Aplicacion, Ventana, and Tipo. Two results are shown, each with a detailed view below it. The first result is for 'Microsoft PowerPoint' with the window title 'Presentación1 - PowerPoint' and type 'keystroke'. The second result is for 'Google Chrome' with the window title 'Tareas - Google Chrome' and type 'keystroke'. Both results show the client name 'alexi' and computer name 'DESKTOP-6723L9T'. The detailed view for each result includes the search term 'los redes neuronales' and 'aula virtual redes epn' respectively, along with buttons for 'AÑADIR A FAVORITOS' and 'ARCHIVAR'.

Fecha	Hora	Aplicacion	Ventana	Tipo
27/01/2022	17:21:53	Microsoft PowerPoint	Presentación1 - PowerPoint	keystroke
Cliente: alexi Computador: DESKTOP-6723L9T				
los redes neuronales				
★ AÑADIR A FAVORITOS 📁 ARCHIVAR				
27/01/2022	17:21:26	Google Chrome	Tareas - Google Chrome	keystroke
Cliente: alexi Computador: DESKTOP-6723L9T				
aula virtual redes epn				
★ AÑADIR A FAVORITOS 📁 ARCHIVAR				

Fig. 34 Resultados de búsqueda exitosa

3.5 *Sprint* 4. Pruebas y despliegue a producción

Los resultados obtenidos al finalizar este *Sprint* han sido los siguientes:

- Pruebas de Aceptación.
- Despliegue de proyecto *backend*.
- Despliegue de proyecto *frontend*.

3.5.1 Pruebas de Aceptación

Una vez que se ha concluido con el desarrollo de las funciones para el *dashboard* web, es importante verificar su funcionamiento. Para esto se ha hecho uso de un recurso comúnmente utilizado en el mundo del desarrollo de software, las pruebas de aceptación de usuario. Estas pruebas tienen el propósito de verificar la satisfacción que el producto de software le produce al usuario final, es decir que, comprueban que el usuario está conforme con el funcionamiento del producto en base a los requerimientos establecidos dentro de escenarios de uso propuestos por el mismo [44].

La **TABLA IV** representa la prueba de aceptación realizada para la funcionalidad de inicio de sesión. El resto de estas pruebas se encuentran en el ANEXO II sección

Pruebas de Aceptación.

TABLA IV. Prueba de Aceptación Nro. 1 - Inicio de sesión

Prueba de Aceptación	
Identificador: PA01	Identificador Historia de Usuario: HU01
Nombre de Prueba de Aceptación: Inicio de sesión	
Descripción: El administrador del dashboard web tiene un perfil de usuario y puede iniciar sesión con este.	
Pasos de ejecución: <ul style="list-style-type: none">• Acceder a la aplicación en el navegador• Llenar los campos del formulario con las credenciales (correo y contraseña)• Pulsar la tecla <i>enter</i> o dar clic en el botón "Ingresar"	
Resultado deseado: Se envían los datos por medio del formulario hacia el API, esta se encarga de validar la concordancia de las credenciales ingresadas con las almacenadas en la base de datos y retorna un token para la sesión del usuario administrador. Se redirige al usuario a la página inicial del dashboard.	
Evaluación de prueba: Resultado exitoso. El <i>dashboard</i> web permite iniciar sesión con el ingreso de las credenciales provistas y redirecciona al usuario a la página principal de este. Aprobación del cliente en un 100%.	

3.5.2 Despliegue de proyecto *backend*

El *backend* del dashboard está compuesto por el API Rest y la base de datos, para alojar a cada uno de estos se ha utilizado los servicios en la nube provistos por la plataforma Heroku [45]. El primer paso que se lleva a cabo para esto es conectar el repositorio del código fuente, alojado en GitHub, con la aplicación creada en Heroku. Esto facilita el despliegue ya que la plataforma se encarga de identificar las ramas que contiene el repositorio y permite al usuario elegir desde cuál de estas desplegar la aplicación. La **Fig. 35** muestra que la aplicación cuenta con un servicio de base de datos propio de la plataforma, Heroku Postgres.

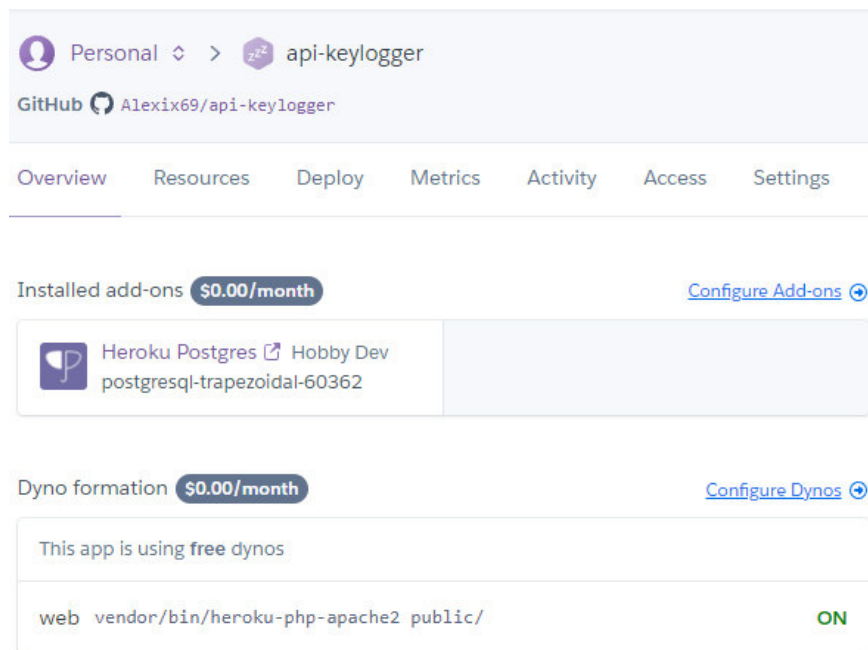


Fig. 35 API Rest desplegada en Heroku

Este componente se ha añadido para almacenar los datos provenientes del *keylogger*, cuenta con su propia configuración junto con credenciales que se deben establecer como variables de entorno del API. Además, se observa que se debe definir el servidor de Apache como público para que el contenido pueda ser accedido una vez que la aplicación se encuentre en producción. La **Fig. 36** muestra las variables de entorno establecidas, entre ellas constan las credenciales para el funcionamiento con la base de datos de Postgres.

Config Vars		Hide Config Vars
APP_DEBUG	true	✎ ✕
APP_ENV	production	✎ ✕
APP_KEY	base64:PKcckXH+PyF2jgUMAn6j127fh8NMgjuQf	✎ ✕
APP_NAME	API_Keylogger	✎ ✕
APP_URL	https://api-keylogger.herokuapp.com/	✎ ✕
DATABASE_URL	postgres://icd1ftofjlzjto:a2d35efeb341c	✎ ✕
DB_CONNECTION	pgsql	✎ ✕
DB_DATABASE	daft6fj8ue12j8	✎ ✕
DB_HOST	ec2-54-157-15-228.compute-1.amazonaws.c	✎ ✕
DB_PASSWORD	a2d35efeb341c37b946ad5249df891865da061e	✎ ✕
DB_PORT	5432	✎ ✕
DB_USERNAME	icd1ftofjlzjto	✎ ✕

Fig. 36 Variables de entorno del API Rest

Una vez realizadas estas configuraciones, el *backend* del *dashboard* web se despliega y se accede a él por medio del dominio público que brinda el servicio de *hosting* gratuito de Heroku. El último paso necesario es ejecutar las migraciones dentro de la nueva base de datos por medio de la consola de la aplicación, esto crea las entidades y relaciones establecidas para almacenar los registros obtenidos por el *keylogger*. Esto lo muestra a continuación la **Fig. 37**.

```

Console

Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (70.44ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (30.79ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (53.12ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (44.40ms)
Migrating: 2021_12_13_174454_create_clients_table
Migrated: 2021_12_13_174454_create_clients_table (28.40ms)
Migrating: 2021_12_14_184252_create_records_table
Migrated: 2021_12_14_184252_create_records_table (44.86ms)
Migrating: 2021_12_14_214914_create_favorite_categories_table
Migrated: 2021_12_14_214914_create_favorite_categories_table (31.65ms)
Migrating: 2021_12_23_074823_add_client_id_column_record
Migrated: 2021_12_23_074823_add_client_id_column_record (32.84ms)
Migrating: 2021_12_23_075610_add_category_id_column_record
Migrated: 2021_12_23_075610_add_category_id_column_record (26.93ms)

```

Fig. 37 Ejecución de migraciones en la base de datos

3.5.3 Despliegue de proyecto *frontend*

Desplegar el proyecto *frontend* del *dashboard* requiere de un proceso similar, para esto se ha hecho uso de la plataforma en la nube Vercel [46]. Al igual que en el despliegue del *backend*, es necesario conectar el repositorio del código fuente al nuevo proyecto creado en la plataforma, esto facilita la configuración de la rama sobre la cual se establece el despliegue a producción. La **Fig. 38** muestra lo mencionado.

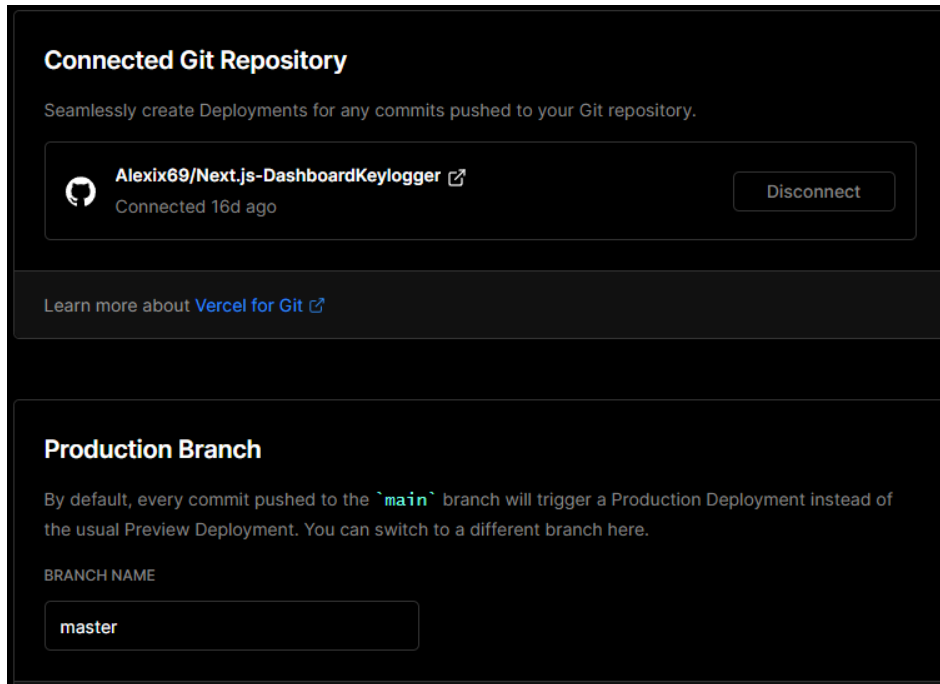


Fig. 38 Conexión de proyecto *frontend* con Vercel

Para que el *frontend* interactúe con el *backend* desplegado se establece el dominio público del proyecto de Heroku como una variable de entorno en el proyecto de Vercel, esto hará que la interfaz de usuario, el API y la base de datos trabajen en conjunto como una sola aplicación. A continuación, se presenta esta configuración en la **Fig. 39**.

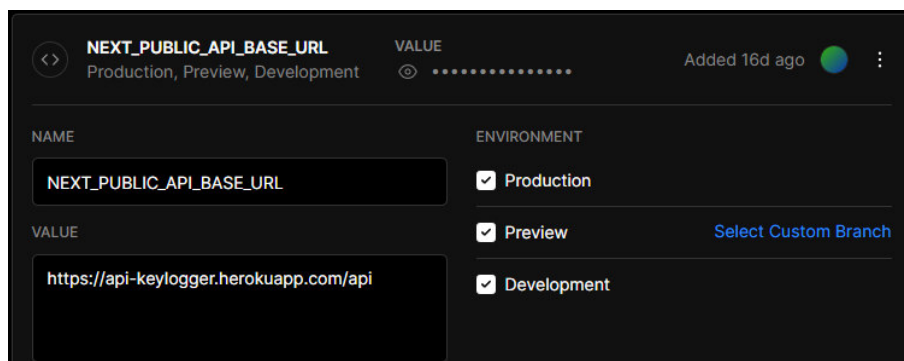


Fig. 39 Variable de entorno del proyecto *frontend*

Concluido este proceso, se puede acceder al *dashboard* web por medio del dominio público de la aplicación provisto por Vercel: [Dashboard Web - Sign In \(dashboard-keylogger.vercel.app\)](https://dashboard-keylogger.vercel.app).

4 CONCLUSIONES

- El desarrollo del *dashboard* web se ha llevado a cabo conforme a los requerimientos planteados al inicio del proyecto, esto ha permitido que se obtenga un producto de calidad que satisface las necesidades del cliente; el usuario administrador.
- El diseño de la base de datos ha requerido de varios cambios importantes durante el desarrollo del proyecto, ya que conforme este avanzaba se encontraron situaciones en las que el diseño inicial no era el óptimo para el flujo de datos, por lo cual, se ha debido normalizar las tablas y relaciones existentes en busca de conseguir eficiencia en las transacciones de datos presentes en el funcionamiento de la aplicación.
- La definición del patrón arquitectónico MVC ha provisto varias ventajas en cuanto al manejo y la presentación de los datos que se establecen en el *dashboard* web, esto gracias a la independización que existe entre cada capa frente a la otra. Este hecho ha permitido la utilización de varias herramientas y tecnologías dedicadas a cumplir propósitos específicos de cada componente de la aplicación.
- La implementación de la metodología Scrum para encaminar el desarrollo de este proyecto ha sido una decisión acertada, ya que durante el transcurso de este se han presentado una serie de cambios imprevistos pero necesarios. Estos fueron rápidamente adaptados y no influyeron drásticamente en la entrega de cada iteración gracias al manejo de artefactos y técnicas propias de esta metodología ágil.
- Los marcos de trabajo usados en este proyecto han sido seleccionados en base al análisis de sus prestaciones y servicios, en el desarrollo del *backend*, Laravel ha cumplido un papel importante ya que se ha adaptado perfectamente a la arquitectura MVC planteada en este proyecto. Por otra parte, Next junto con React han servido para la construcción de una interfaz gráfica eficiente que permite la interacción del usuario con los datos que en esta se presentan.
- La verificación del funcionamiento es un proceso por el cual ha debido pasar el *dashboard* web, se efectuaron las respectivas pruebas de aceptación ya que este tipo de sistema tiene el enfoque de ser usado por un solo tipo de usuario; el administrador. Este proceso ha evaluado el nivel de satisfacción producida en el usuario mientras hace uso de la aplicación, la aceptación de este ha sido afirmativa en la implementación de la mayoría de los requisitos propuestos en un inicio.

- Desplegar una aplicación a producción es un proceso que, entre varios aspectos, principalmente involucra inversión económica. Este es un punto importante que se considera durante el desarrollo de un producto de software, sin embargo, en busca de verificar el comportamiento del *dashboard* web en un entorno de producción, se ha hecho uso de planes gratuitos disponibles en las plataformas elegidas para esto, Heroku para el *backend* y Vercel para el *frontend*.

5 RECOMENDACIONES

- Se recomienda que, para mantener un flujo de desarrollo eficiente dentro de un proyecto de software, se defina una metodología sobre la cual se va a trabajar. Esto de ninguna manera eliminará todos los inconvenientes y problemas que se presentan en los procesos, sin embargo, si hará una gran diferencia en cuanto a la resolución y manejo que se le pueda dar a estos eventos. Es importante mencionar también, que la capacidad de respuesta será proporcional al nivel de diseño y planificación previo que se haya realizado antes de dar inicio al proyecto.
- Es importante que en la medida de lo posible se posicione a la experiencia de usuario delante de otros aspectos que conforman un sistema web. Esto quiere decir que, en cada proceso, funcionalidad nueva, adaptación de hardware, entre otros factores, se priorice la usabilidad que se le ofrecerá al usuario después de implementar un cambio. Esto en la mayoría de los casos garantiza la aprobación del sistema por el usuario final, sin embargo, siempre hay que identificar el punto de equilibrio entre la calidad que se va a ofrecer y los recursos disponibles.
- La escalabilidad de un sistema se puede obtener por medio de varios factores, uno de los más comunes y productivos, es la separación de componentes y responsabilidades. En este contexto, se recomienda que se analice a fondo el tipo de sistema a construir antes de implementar su arquitectura, entre varios aspectos se puede considerar; el número de usuarios de la aplicación, el volumen de datos que se va a manejar, la disponibilidad que esta le ofrece al usuario final, entre otros. El construir una aplicación distribuida no siempre será la alternativa correcta ni la más fácil de implementar, no obstante, si se busca escalabilidad, es una elección conveniente.
- Es común que ante un problema o error presentado durante la codificación e implementación de alguna nueva herramienta se busque la solución en foros o sitios de Internet relacionados con la informática, esto puede resultar útil en ocasiones y resolver cierta cantidad de problemas, sin embargo, se recomienda que ante este

tipo de eventos la primera acción a llevar a cabo sea el acceder a la documentación oficial de la herramienta o tecnología que se esté usando. Los beneficios que esto conlleva son claros; se garantiza el aprendizaje del manejo de la herramienta y el desarrollador adquiere nuevas habilidades para solventar el problema actual y aquellos que podrían llegar a presentarse en el futuro.

- Se recomienda actualizar las versiones de herramientas, librerías y tecnologías a las más actuales. El probar nuevas funcionalidades o aditamentos de una tecnología emergente solo se lo debe hacer en un ambiente de desarrollo, nunca en producción. Es importante considerar que cada proyecto de software es independiente y en ocasiones no se va a requerir la versión más actual de una herramienta sino la más estable, por ello hay que tener en cuenta en que contexto es conveniente actualizar o no una herramienta, librería o tecnología.
- Para finalizar, se recomienda acceder a los servicios pagados de una plataforma si se desea obtener escalabilidad y potenciación a nivel profesional de una aplicación en producción, ya que las prestaciones gratuitas de la mayoría de estas brindan pocos recursos o están diseñadas para proveer funcionalidades y servicios de uso básico. Una alternativa a esto es usar herramientas basadas en código abierto, pero por supuesto, su implementación y mantenimiento requerirán de un mayor conocimiento.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] R. P. Rivera Guevara, «Análisis de características estáticas de ficheros ejecutables para la clasificación de malware,» Tesis de Máster, Universidad Politécnica de Madrid, España, Madrid, 2014.
- [2] Anónimo, «PHP - Wikipedia, la enciclopedia libre,» 14 Enero 2022. [En línea]. Available: <https://es.wikipedia.org/wiki/PHP>. [Último acceso: 14 Febrero 2022].
- [3] R. Ramos, «¿Qué es JavaScript y para qué sirve?,» Agencia de Marketing Rafa Ramos, 2016. [En línea]. Available: <https://soyrafaramos.com/que-es-javascript-para-que-sirve/>. [Último acceso: 14 Febrero 2022].
- [4] R. P. Rivera-Guevara, «Deteccion y clasificacion de malware con el sistema de análisis de malware cuckoo.,» UNIR, Master's thesis, 2018.
- [5] P. U. Mejías, «Material UI / Caching en React - TechClub Tajamar,» 22 Junio 2020. [En línea]. Available: <https://techclub.tajamar.es/material-ui-caching-en-react/>. [Último acceso: 15 Febrero 2022].
- [6] A. C. Urbano, «Styled Components Basics (Spanish) - DEV Comunity,» 31 Mayo 2020. [En línea]. Available: <https://dev.to/alfredocu/styled-components-basics-spanish-jdn>. [Último acceso: 15 Febrero 2022].
- [7] Rootstack, «Características más poderosas de Laravel | Rootstack,» 24 Septiembre 2021. [En línea]. Available: <https://www.rootstack.com/es/blog/laravel-caracteristicas-framework/>. [Último acceso: 15 Febrero 2022].
- [8] A. B. Martínez, «¿Qué es un ORM y para qué sirve Eloquent - Platzi?,» Febrero 2017. [En línea]. Available: <https://platzi.com/clases/1920-eloquent-laravel/28515-que-es-un-orm-y-para-que-sirve-eloquent/>. [Último acceso: 15 Febrero 2022].
- [9] S. Borges, «¿Qué es PostgreSQL? - Para qué sirve, Características e Instalación,» 19 Noviembre 2019. [En línea]. Available: <https://blog.infranetworking.com/servidor-postgresql/>. [Último acceso: 15 Febrero 2022].
- [10] R. Rivera, P. Kotzias, A. Sudhodanan y J. Caballero, «Costly freeware: a systematic analysis of abuse in download portals.,» *IET Information Security*, 13(1), pp. 27-35, 2019.
- [11] P. Kotzias, S. Matic, R. Rivera y J. Caballero, «Certified PUP: abuse in authenticode code signing.,» *In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security.*, pp. 465-478, 2015.
- [12] C. Jiménez y R. Rivera, «Ciberseguridad del IoT: Un Análisis en Países de la Unión Europea,» *Revista Ibérica de Sistemas e Tecnologías de Informação*, nº 39, pp. 461-476, 2021.
- [13] L. Pazmiño, F. Flores, L. Ponce, J. Zaldumbide, V. Parraga, B. Loarte, G. Cevallos, I. Maldonado and R. Rivera, "Challenges and Opportunities of IoT Deployment in Ecuador," in 2019

International Conference on Information Systems and Software Technologies (ICI2ST), Quito, 2019.

- [14] M. Sebastián, R. Rivera, P. Kotzias y J. Caballero, «Avclass: A tool for massive malware labeling,» de *International symposium on research in attacks, intrusions, and defenses*, 2016.
- [15] R. Rivera, L. Pazmiño, F. Becerra y J. Barriga, «An Analysis of Cyber Espionage Process,» de *Developments and Advances in Defense and Security. Proceedings of MICRADS 2021*, Cartagena, 2021.
- [16] J. F. Pareja Quinaluisa, «Evaluación de procesos de software utilizando EvalProSoft Aplicado a un caso de estudio,» 08 02 2012. [En línea]. Available: <http://bibdigital.epn.edu.ec/handle/15000/4491> .
- [17] Santander Universidades, «Metodologías de desarrollo software | Blog Becas Santander,» 21 Diciembre 2020. [En línea]. Available: <https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html>. [Último acceso: 15 Febrero 2022].
- [18] P. Stefaniak, «¿Qué es Backend y Frontend? - Descubre Comunicación,» 26 Julio 2019. [En línea]. Available: <https://descubrecomunicacion.com/que-es-backend-y-frontend/>. [Último acceso: 15 Febrero 2022].
- [19] QuestionPro, «¿Qué es un estudio de caso y cómo realizarlo?,» [En línea]. Available: <https://www.questionpro.com/blog/es/que-es-un-estudio-de-caso/>. [Último acceso: 15 Febrero 2022].
- [20] M. F. Aguirre, «Prueba de Concepto (PoC): qué es y ejemplo de su utilidad,» 26 Enero 2021. [En línea]. Available: Prueba de Concepto (PoC): qué es y ejemplo de su utilidad. [Último acceso: 15 Febrero 2022].
- [21] Anónimo, «Qué es SCRUM – Proyectos Ágiles,» [En línea]. Available: <https://proyectosagiles.org/que-es-scrum/>. [Último acceso: 15 Febrero 2022].
- [22] A. R. Mesa, «Qué es un Sprint de Scrum | OpenWebinars,» 19 Diciembre 2018. [En línea]. Available: <https://openwebinars.net/blog/que-es-un-sprint-scrum/>. [Último acceso: 15 Febrero 2022].
- [23] Be Agile My Friend, «Product Owner, Qué es y Cuáles son sus funciones. Cómo ser el Mejor!,» Be Agile My Friend, [En línea]. Available: https://beagilemyfriend.com/product-owner/#%C2%BFQue_es_un_Product_Owner. [Último acceso: 16 Febrero 2022].
- [24] P. CANAL, «Definición y características del Scrum Master,» 10 Septiembre 2015. [En línea]. Available: <https://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/>. [Último acceso: 15 Febrero 2022].
- [25] J. Roche, «Artefactos Scrum: las 3 herramientas clave de gestión,» [En línea]. Available: <https://www2.deloitte.com/es/es/pages/technology/articles/artefactos-scrum.html>. [Último acceso: 16 Febrero 2022].

- [26] M. REHKOPF, «Historias de usuario | Ejemplos y plantilla | Atlassian,» [En línea]. Available: <https://www.atlassian.com/es/agile/project-management/user-stories>. [Último acceso: 15 Febrero 2022].
- [27] K. Bracey, «¿Qué es Figma?,» 26 Noviembre 2018. [En línea]. Available: <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>. [Último acceso: 15 Febrero 2022].
- [28] Anónimo, «Software User Interface Design,» [En línea]. Available: https://www.tutorialspoint.com/software_engineering/software_user_interface_design.htm. [Último acceso: 15 Febrero 2022].
- [29] Interaction Design FOUNDATION, «What is User Experience (UX) Design? | Interaction Design Foundation (IxDF),» [En línea]. Available: <https://www.interaction-design.org/literature/topics/ux-design>. [Último acceso: 15 Febrero 2022].
- [30] Servicio de Informática ASP.NET MVC 3 Framework, «Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework,» Servicio de Informática ASP.NET, [En línea]. Available: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. [Último acceso: 15 Febrero 2022].
- [31] M. Garcia, «¿QUE ES XAMPP Y COMO PUEDO USARLO? | Nettix Perú,» 30 Mayo 2020. [En línea]. Available: <https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo/>. [Último acceso: 15 Febrero 2022].
- [32] A. López, «Qué es Postman y para qué sirve | OpenWebinars,» 3 Junio 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-postman/>. [Último acceso: 16 Febrero 2022].
- [33] JetBrains, «Funcionalidades - PhpStorm,» JetBrains, [En línea]. Available: <https://www.jetbrains.com/es-es/phpstorm/features/>. [Último acceso: 15 Febrero 2022].
- [34] J. Lucas, «Qué es NodeJS y para qué sirve | OpenWebinars,» 4 Septiembre 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-nodejs/>. [Último acceso: 16 Febrero 2022].
- [35] COR, «¿Next.js es lo Mismo que React.js? - COR,» [En línea]. Available: <https://projectcor.com/es/blog/next-js-es-lo-mismo-que-react-js/>. [Último acceso: 15 Febrero 2022].
- [36] J. L. COALLA, «React | Qué es, para qué sirve y cómo funciona | Descúbrelo todo,» 23 Junio 2021. [En línea]. Available: <https://tech.tribalyte.eu/blog-que-es-react>. [Último acceso: 15 Febrero 2022].
- [37] M. Shokeen, «Comenzando Con Chart.js: Introducción,» 18 Abril 2017. [En línea]. Available: <https://code.tutsplus.com/es/tutorials/getting-started-with-chartjs-introduction--cms-28278>. [Último acceso: 15 Febrero 2022].

- [38] JetBrains, «WebStorm: funcionalidades,» JetBrains, [En línea]. Available: <https://www.jetbrains.com/es-es/webstorm/features/#specialized-in-javascript>. [Último acceso: 15 Febrero 2022].
- [39] G. B., «GitHub: ¿Qué Es GitHub Y Cómo Utilizarlo?,» 8 Marzo 2021. [En línea]. Available: <https://www.hostinger.es/tutoriales/que-es-github>. [Último acceso: 15 Febrero 2022].
- [40] Yair, «Qué es Composer y cómo usarlo – Styde.net,» 23 Diciembre 2019. [En línea]. Available: <https://styde.net/que-es-composer-y-como-usarlo/>. [Último acceso: 15 Febrero 2022].
- [41] jquense, «jquense/yup: Dead simple Object schema validation,» 24 Junio 2016. [En línea]. Available: <https://github.com/jquense/yup>. [Último acceso: 15 Febrero 2022].
- [42] R. R. Guevara, Tools for the detection and analysis of potentially unwanted programs, (Doctoral dissertation, Tesis doct. Nov. de 2018. doi: 10.20868/UPM.thesis.53395), 2018.
- [43] Vercel, «Basic Features: Image Optimization | Next.js,» Vercel, [En línea]. Available: <https://nextjs.org/docs/basic-features/image-optimization>. [Último acceso: 15 Febrero 2022].
- [44] PMOinformatica.com, «Pruebas de aceptación de software según el ISTQB,» 8 Agosto 2016. [En línea]. Available: <http://www.pmoinformatica.com/2016/08/pruebas-aceptacion-software-istqb.html>. [Último acceso: 15 Febrero 2022].
- [45] ricardocelis, «Heroku: qué es, cómo funciona y para qué sirve - Platzi,» 2018. [En línea]. Available: <https://platzi.com/blog/que-es-heroku/>. [Último acceso: 15 Febrero 2022].
- [46] iProUP, «Argentina tiene sexto unicornio argentino: qué es Vercel,» 23 Junio 2021. [En línea]. Available: <https://www.iproup.com/startups/23899-argentina-tiene-sexto-unicornio-argentino-que-es-vercel>. [Último acceso: 15 Febrero 2022].

7 ANEXOS

ANEXO I. Certificado de originalidad

ANEXO II. Manual técnico

ANEXO III. Manual de usuario

ANEXO III. Manual de instalación

Anexo I. Certificado de Originalidad

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 24 de febrero de 2022

De mi consideración:

Yo, Richard Paúl Rivera Guevara, en calidad de director del Trabajo de Integración Curricular titulado DESARROLLO DE UN DASHBOARD WEB asociado al proyecto DESARROLLO DE UN KEYLOGGER PARA WINDOWS 11 CON UN DASHBOARD WEB, elaborado por el estudiante ALEXIS FERNANDO YÉPEZ FUEL de la carrera en TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo, producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 8%.

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 en el Quipux correspondiente.

Atentamente,

**RICHARD
PAUL
RIVERA
GUEVARA** Firmado
digitalmente por
RICHARD PAUL
RIVERA GUEVARA
Fecha: 2022.02.24
12:20:04 -05'00'

Dr. Richard Rivera.
Profesor EPN-ESFOT

ANEXO II. Manual Técnico

Recopilación de requerimientos

La **TABLA V** presenta los requerimientos obtenidos para el desarrollo del *dashboard* web.

TABLA V. Recopilación de requerimientos

RECOPIACIÓN DE REQUERIMIENTOS	
ID-RR	ENUNCIADO DEL ÍTEM
RR01	Como usuario administrador del <i>dashboard</i> , necesito tener un perfil para iniciar sesión.
RR02	Como usuario administrador del <i>dashboard</i> , necesito visualizar las pulsaciones realizadas en el teclado.
RR03	Como usuario administrador del <i>dashboard</i> , necesito visualizar las capturas de pantalla realizadas por el <i>keylogger</i> .
RR04	Como usuario administrador del <i>dashboard</i> , necesito visualizar las pulsaciones de teclas realizadas en sitios web que han sido visitados por el cliente.
RR05	Como usuario administrador del <i>dashboard</i> , necesito marcar a un registro como favorito y tener acceso a él.
RR06	Como usuario administrador del <i>dashboard</i> , necesito visualizar el listado de clientes junto con el total de registros captados de cada tipo.
RR07	Como usuario administrador del <i>dashboard</i> , necesito ordenar los registros según el nombre de aplicación, título de ventana, fecha y hora.
RR08	Como usuario administrador del <i>dashboard</i> , necesito buscar registros por nombre de aplicación, título de ventana, fecha y hora.
RR09	Como usuario administrador del <i>dashboard</i> , necesito visualizar la lista de registros con propiedades como la hora, fecha y nombre de la aplicación en donde fueron generados.
RR10	Como usuario del <i>dashboard</i> , necesito archivar la información que considere innecesaria.
RR11	Como usuario administrador del <i>dashboard</i> , necesito crear categorías para clasificar registros importantes.
RR12	Como usuario administrador del <i>dashboard</i> , necesito un buscador para encontrar palabras dentro de las pulsaciones del teclado registradas.
RR13	Como usuario del <i>dashboard</i> , necesito descargar las capturas de pantalla que se hayan registrado.
RR14	Como usuario administrador del <i>dashboard</i> , necesito visualizar un gráfico estadístico que muestre el número total de registros por cada cliente.
RR15	Como usuario del <i>dashboard</i> , necesito visualizar un gráfico estadístico de cada tipo de registro que muestre el número total de estos por cada cliente.
RR16	Como usuario administrador del <i>dashboard</i> , necesito agregar y eliminar registros de una categoría.
RR17	Como usuario administrador del <i>dashboard</i> , necesito administrar las categorías junto con los registros que pertenezcan a cada una.
RR18	Como usuario administrador del <i>dashboard</i> , necesito filtrar registros según su tipo.

Historias de Usuario

A continuación, se muestran las Historias de usuario creadas para este proyecto. El propósito que tienen y un ejemplo de estas se ha detallado en la sección Artefactos. Desde la **TABLA VI** hasta la **TABLA XXII** se presentan las Historias de usuario restantes.

TABLA VI. Historia de usuario Nro. 2 - Visualización de pulsaciones

HISTORIA DE USUARIO	
Identificador (ID): HU02	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Visualización de pulsaciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar, en un área específica de la pantalla, las pulsaciones realizadas en el equipo infectado con el <i>keylogger</i> .	
Observación: El usuario podrá visualizar las pulsaciones realizadas en cada registro generado por el <i>keylogger</i> .	

TABLA VII. Historia de usuario Nro. 3 - Visualización de capturas de pantalla

HISTORIA DE USUARIO	
Identificador (ID): HU03	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Visualización de capturas de pantalla	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar, en un área específica de la pantalla, las capturas de pantalla realizadas en el equipo infectado con el <i>keylogger</i> .	
Observación: El usuario podrá visualizar las capturas de pantalla realizadas en cada registro generado por el <i>keylogger</i> .	

TABLA VIII. Historia de usuario Nro. 4 - Visualización de pulsaciones en sitios web

HISTORIA DE USUARIO	
Identificador (ID): HU04	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Visualización de pulsaciones en sitios web	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar, en un área específica de la pantalla, las pulsaciones de teclas realizadas en los sitios web visitados desde el equipo infectado con el <i>keylogger</i> .	
Observación: El usuario tendrá acceso a la información de los sitios web visitados.	

TABLA IX. Historia de usuario Nro. 5 - Marcar y desmarcar un registro como favorito

HISTORIA DE USUARIO	
Identificador (ID): HU05	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Marcar y desmarcar un registro como favorito	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 3	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá marcar un registro como favorito, también podrá quitarlo de ahí cuando crea necesario.	
Observación: El usuario podrá agregar un registro a la sección de favoritos, podrá acceder en una ventana específica y también eliminarlo de allí cuando desee.	

TABLA X. Historia de usuario Nro. 6 - Visualizar listado de clientes

HISTORIA DE USUARIO	
Identificador (ID): HU06	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Visualizar listado de clientes	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Iteración Asignada: 2	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar el listado de clientes en una ventana específica, así como también un detalle del número total de registros que han sido captados de este.	
Observación: El usuario podrá visualizar el número de registros totales captados de un cliente y el número de estos según el tipo al que pertenecen.	

TABLA XI. Historia de usuario Nro. 7 - Organización de información por parámetros

HISTORIA DE USUARIO	
Identificador (ID): HU07	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Organización de información por parámetros	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Iteración Asignada: 2	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá organizar los registros (pulsaciones, capturas de pantalla, sitios web visitados) de acuerdo con un parámetro específico de estos.	
Observación: El usuario podrá cambiar el orden de presentación de los registros de acuerdo con la fecha, hora, título de ventana y nombre de aplicación según lo requiera.	

TABLA XII. Historia de usuario Nro. 8 - Búsqueda de registros por parámetros

HISTORIA DE USUARIO	
Identificador (ID): HU08	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Búsqueda de registros por parámetros	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá localizar registros de manera rápida por medio de la búsqueda de un parámetro específico de estos.	
Observación: El usuario dispondrá de un filtro que devuelva los registros que coincidan con el parámetro buscado, los parámetros disponibles serán la fecha, la hora, título de ventana y nombre de aplicación.	

TABLA XIII. Historia de usuario Nro. 9 - Visualización de registros con parámetros

HISTORIA DE USUARIO	
Identificador (ID): HU09	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Visualización de registros con parámetros	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar, en una ventana específica, la lista de los registros generados por el <i>keylogger</i> con propiedades de cada uno de estos como; la hora, fecha, título de ventana y nombre de la aplicación en la que hayan sido obtenidos.	
Observación: El usuario verá la lista de registros en una tabla que tenga como columnas los parámetros antes mencionados, pertenecientes a cada registro generado por el <i>keylogger</i> .	

TABLA XIV. Historia de usuario Nro. 10 - Archivar y desarchivar información

HISTORIA DE USUARIO	
Identificador (ID): HU10	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Archivar y desarchivar información	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Iteración Asignada: 3	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá archivar la información que considere innecesaria (pulsaciones, capturas de pantalla, sitios web visitados). Esta sección contendrá aquellos registros que hayan sido marcados como archivados y podrá quitar de ahí un registro específico cuando lo desee.	
Observación: El usuario tendrá un apartado dentro del <i>dashboard</i> web, el cual le permitirá acceder a la información que haya archivado con anticipación y contará con la opción de desarchivar un registro cuando lo desee.	

TABLA XV. Historia de usuario Nro. 11 - Crear categorías

HISTORIA DE USUARIO	
Identificador (ID): HU11	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Crear categorías	
Prioridad en negocio: Baja	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web tendrá la opción de crear categorías para poder clasificar registros de mejor manera.	
Observación: El usuario tendrá un apartado dentro del <i>dashboard</i> web, en el cual ingresará el nombre y una descripción (opcional) para crear una categoría de registros favoritos.	

TABLA XVI. Historia de usuario Nro. 12 - Buscador de palabras

HISTORIA DE USUARIO	
Identificador (ID): HU12	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Buscador de palabras	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web dispondrá de un buscador del cual podrá hacer uso para encontrar palabras específicas dentro de los registros de pulsaciones de teclas generados por el <i>keylogger</i> .	
Observación: El buscador arrojará diferentes tipos de resultados dependiendo de la exactitud que tenga la búsqueda que realice el usuario y la coincidencia que esta tenga con el contenido de cada registro.	

TABLA XVII. Historia de usuario Nro. 13 - Descarga de imágenes

HISTORIA DE USUARIO	
Identificador (ID): HU13	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Descarga de imágenes	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Iteración Asignada: 3	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá descargar las capturas de pantalla registradas por el <i>keylogger</i> que desee, esto por medio de un botón para posteriormente almacenarlas de forma local.	
Observación: La descarga de imágenes permitirá al usuario tener acceso a estas aun cuando no esté usando el <i>dashboard</i> web, ya que éstas se encontrarán almacenadas localmente.	

TABLA XVIII. Historia de usuario Nro. 14 - Gráfico estadístico registros-clientes

HISTORIA DE USUARIO	
Identificador (ID): HU14	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Gráfico estadístico registros-clientes	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Iteración Asignada: 2	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar un gráfico estadístico de tipo dona en el que se represente el número de registros obtenidos por cada cliente.	
Observación: El gráfico estadístico estará basado en los datos arrojados de los registros y ofrecerá un resumen en tiempo real.	

TABLA XIX. Historia de usuario Nro. 15 - Gráfico estadístico de tipos de registros

HISTORIA DE USUARIO	
Identificador (ID): HU15	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Gráfico estadístico de tipos de registros	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Iteración Asignada: 3	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá visualizar un gráfico estadístico de tipo pastel en el que se represente la cantidad de registros dependiendo del tipo que sean y en relación con el cliente del cual hayan sido obtenidos.	
Observación: El gráfico estadístico estará basado en el número de registros totales, pero clasificados de acuerdo con el tipo de cada uno. De igual manera, estarán relacionados con cada cliente.	

TABLA XX. Historia de usuario Nro. 16 - Agregar registros a una categoría

HISTORIA DE USUARIO	
Identificador (ID): HU16	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Agregar registros a una categoría	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Iteración Asignada: 3	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web podrá elegir un registro y agregarlo a una categoría para tener acceso a él dentro de la sección de favoritos. Si así lo considera, también podrá eliminarlo de la categoría a la que pertenezca.	
Observación: Para visualizar los registros que pertenecen a una categoría, el usuario podrá acceder a la sección de favoritos.	

TABLA XXI. Historia de usuario Nro. 17 - Administrar categorías

HISTORIA DE USUARIO	
Identificador (ID): HU17	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Administrar categorías	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Iteración Asignada: 2	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web dispondrá de un selector para escoger entre las categorías existentes, este a su vez le permitirá desplegar los registros pertenecientes a cada categoría.	
Observación: El selector de categorías se encontrará en la ventana de favoritos dentro del <i>dashboard</i> , esto ya que las categorías están enfocadas a clasificar de mejor manera los registros considerados más importantes.	

TABLA XXII. Historia de usuario Nro. 18 - Filtro de registros por tipo

HISTORIA DE USUARIO	
Identificador (ID): HU18	Usuario: Administrador del <i>Dashboard</i>
Nombre Historia: Filtro de registros por tipo	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Iteración Asignada: 1	
Responsable: Alexis Yépez	
Descripción: El administrador del <i>dashboard</i> web dispondrá, dentro del listado principal de todos los registros, la funcionalidad de filtrarlos de acuerdo con el tipo de cada uno.	
Observación: El usuario tendrá la opción de filtrar los registros según su tipo, es decir, capturas de pantalla, pulsaciones de teclas y sitios web visitados.	

Product Backlog

La **TABLA XXIII** presenta el *Product Backlog* definido para el desarrollo del *dashboard* web, este contiene la priorización, estado y número de *Sprint* en el que se ha llevado a cabo el cumplimiento de cada Historia de usuario.

TABLA XXIII. *Product Backlog*

ELABORACIÓN DEL <i>PRODUCT BACKLOG</i>				
ID-HU	HISTORIA DE USUARIO	ITERACIÓN	ESTADO	PRIORIDAD
HU01	Perfil de usuario e inicio de sesión	1	Finalizada	Alta
HU02	Visualización de pulsaciones realizadas	1	Finalizada	Alta
HU03	Visualización de capturas de pantalla	1	Finalizada	Alta
HU04	Visualización de sitios web visitados	1	Finalizada	Alta
HU05	Marcar y desmarcar un registro como favorito	3	Finalizada	Media
HU06	Visualizar listado de clientes	2	Finalizada	Media
HU07	Organización de información por parámetros	2	Finalizada	Baja
HU08	Búsqueda de registros por parámetros	1	Finalizada	Media
HU09	Visualización de registros con parámetros	1	Finalizada	Alta
HU10	Archivar y desarchivar información	3	Finalizada	Media
HU11	Crear categorías	2	Finalizada	Baja
HU12	Buscador de palabras	2	Finalizada	Media
HU13	Descarga de imágenes	3	Finalizada	Media
HU14	Gráfico estadístico registros-clientes	2	Finalizada	Media
HU15	Gráfico estadístico de tipos de registros	3	Finalizada	Baja
HU16	Agregar registros a categoría	3	Finalizada	Baja
HU17	Administrar categorías	2	Finalizada	Media
HU18	Filtro de registros por tipo	1	Finalizada	Alta

Sprint Backlog

La **TABLA XXIV** muestra el *Sprint Backlog* utilizado en el proyecto, la importancia que tiene este recurso se explica en la sección *Artefactos*.

TABLA XXIV. *Sprint Backlog*

ELABORACIÓN DEL <i>SPRINT BACKLOG</i>					
ID-SB	NOMBRE	ID-HU	HISTORIA DE USUARIO	TAREAS	TIEMPO ESTIMADO
SB00	Configuración del entorno de desarrollo	N/A	N/A	<ul style="list-style-type: none"> • Instalación de herramientas de desarrollo. • Diseño y creación de la BDD. • Creación del proyecto en Laravel. • Creación del proyecto en Next. 	10H
SB01	Módulo de presentación de la información	HU01	Perfil de usuario e inicio de sesión	<ul style="list-style-type: none"> • Validar los campos de la entidad de usuario en la BD. • Crear un usuario administrador con sus credenciales en la BD. • Implementar un formulario de inicio de sesión. • Verificar que el usuario creado pueda iniciar sesión y que tenga acceso al contenido del <i>dashboard</i>. 	65H
		HU02	Visualización de pulsaciones realizadas	<ul style="list-style-type: none"> • Implementar en la interfaz un área de texto en la que se visualice las pulsaciones del teclado. • Extraer las pulsaciones del teclado registradas de la BD. • Comprobar la concordancia de las pulsaciones visualizadas con las almacenadas en la BD. • Añadir esta funcionalidad como un apartado en la pantalla principal del <i>dashboard</i>. 	
		HU03	Visualización de capturas de pantalla	<ul style="list-style-type: none"> • Implementar un apartado en la interfaz principal dedicada a la visualización de capturas de pantalla. • Extraer las imágenes almacenadas de la BD. • Comprobar la concordancia de las capturas visualizadas con las almacenadas en la BD. • Añadir esta funcionalidad como un apartado en la pantalla principal del <i>dashboard</i>. 	

		HU04	Visualización de sitios web visitados	<ul style="list-style-type: none"> • Implementar en la interfaz principal, un apartado dedicado a la visualización de las pulsaciones de teclas obtenidas de los sitios web visitados por el usuario. • Extraer el nombre del sitio web perteneciente al registro creado de este. • Comprobar la concordancia de los nombres de los sitios con los almacenados en la BD. • Añadir esta funcionalidad como un apartado en la pantalla principal del <i>dashboard</i>. 	
		HU08	Búsqueda de registros por parámetros específicos	<ul style="list-style-type: none"> • Implementar en la pantalla principal de registros un cuadro de texto para ingresar términos de búsqueda. • Establecer los parámetros de búsqueda: nombre de aplicación, título de ventana, fecha y hora. • Verificar que los registros presentados después de ejecutar la búsqueda sean concordantes con el dato ingresado. 	
		HU09	Visualización de registros con parámetros específicos	<ul style="list-style-type: none"> • Implementar una interfaz para mostrar todos los registros como una lista. • Extraer los registros con sus datos de la BD por medio del API Rest. • Corroborar que los parámetros de cada registro concuerden con los visualizados en la tabla. • Añadir esta funcionalidad como un apartado en el menú principal del <i>dashboard</i>. 	
		HU18	Filtro de registros por tipo	<ul style="list-style-type: none"> • Implementar la funcionalidad de filtrado en el listado principal de los registros. • Definir los tres tipos de filtrado: pulsaciones de teclas, capturas de pantalla y sitios web. • Verificar que el filtrado se efectuó en la lista y ésta presente los registros que concuerden con el tipo especificado. 	
SB02	Módulo de organización de la información	HU06	Visualizar listado de clientes	<ul style="list-style-type: none"> • Implementar una interfaz dedicada a la presentación de todos los clientes. • Codificar la funcionalidad de cálculo y clasificación de registros en el API. • Recuperar la información de cada cliente y presentarla en la interfaz de usuario. • Verificar que la información presentada sea concordante con la almacenada en la base de datos y calculada por el API. 	60H


HU07	Organización de registros por parámetros específicos	<ul style="list-style-type: none"> • Implementar la funcionalidad de ordenamiento por parámetros como: nombre de la aplicación, título de ventana, fecha y hora. • Comprobar que el ordenamiento de los registros concuerde con la necesidad del usuario. • Añadir esta funcionalidad a la tabla general de visualización de registros.
HU11	Crear categorías	<ul style="list-style-type: none"> • Establecer a nivel de base de datos, la tabla de entidad Categorías. • Codificar en el API la funcionalidad de crear nueva categoría. • Implementar en la interfaz principal de registros un botón de acceso al formulario para crear una categoría. • Establecer las respectivas validaciones de datos dentro del formulario. • Verificar el envío de datos y la correcta creación de una categoría a nivel de interfaz y BD.
HU17	Administrar categorías	<ul style="list-style-type: none"> • Implementar en el API la función de recuperación de categorías almacenadas en la BD. • Codificar un componente de tipo selector que contenga las categorías existentes. • Verificar que los datos presentados en el componente sean concordantes con los almacenados en la BD. • Codificar la funcionalidad de presentación de registros pertenecientes a cada categoría. • Verificar que los registros presentados sean correspondientes a la categoría asignada previamente. • Añadir este componente y sus funcionalidades dentro del apartado de favoritos.
HU14	Gráfico estadístico registros-clientes	<ul style="list-style-type: none"> • Implementar librerías de gráficos estadísticos dentro del proyecto front-end. • Implementar un gráfico estadístico de tipo dona que sea alimentado por datos almacenados en la BD. • Comprobar que los datos graficados sean correspondientes a la relación registros-clientes. • Verificar que la representación del gráfico sea actual en todo momento y congruente con la información: número total de registros obtenidos de un cliente específico. • Añadir este gráfico estadístico a la pantalla de inicio el <i>dashboard</i>.

		HU16	Agregar registros a categoría	<ul style="list-style-type: none"> • Implementar en el API la funcionalidad que permita agregar un registro a una categoría. • Codificar en la interfaz de usuario un listado de las categorías existentes. • Establecer las validaciones correspondientes al momento de asignar un registro a una categoría. • Verificar que el registro se haya asignado a la categoría escogida. • Añadir esta funcionalidad en el botón "Añadir a favoritos" de cada registro. 	
SB03	Módulo de servicios adicionales para el usuario	HU05	Marcar y desmarcar un registro como favorito	<ul style="list-style-type: none"> • Implementar un botón que permita enviar el registro deseado al apartado de favoritos a nivel de interfaz y BD. • Implementar el atributo y funcionalidad que establezca a un registro de cualquier tipo como favorito. • Añadir un apartado en el menú principal del <i>dashboard</i> al cual se pueda acceder para visualizar todos los registros que hayan sido marcados como favoritos. • Comprobar el correcto funcionamiento a nivel de interfaz y de BD. • Implementar un botón que permita desmarcar a un registro como favorito a nivel de interfaz y de BD. 	55H
		HU10	Archivar y desarchivar registro	<ul style="list-style-type: none"> • Implementar un botón que permita enviar el registro deseado al apartado de información archivada a nivel de interfaz y de BD. • Implementar el atributo y funcionalidad que establezca a un registro de cualquier tipo como archivado. • Añadir un apartado en el menú principal del <i>dashboard</i> al cual se pueda acceder para visualizar todos los registros que hayan sido marcados como archivados. • Comprobar el archivamiento correcto de los registros. • Implementar un botón que permita desmarcar a un registro como archivado a nivel de interfaz y de BD. 	
		HU13	Descarga de imágenes	<ul style="list-style-type: none"> • Implementar un botón que permita la descarga de imágenes de acuerdo con el registro al que pertenece. • Implementar funcionalidad de descarga de imágenes en el API Rest y a nivel de interfaz. • Comprobar que la descarga de cada imagen se ejecute correctamente. 	

		HU15	Gráfico estadístico de tipos de registros	<ul style="list-style-type: none"> • Implementar librerías de gráficos estadísticos dentro del proyecto front-end. • Implementar un gráfico estadístico de tipo pastel que sea alimentado por datos proveniente de la BD. • Comprobar que los datos graficados sean correspondientes a los almacenados en la BD. • Verificar que la representación del gráfico sea actual en todo momento y congruente con la información: número de registros de cada tipo por cliente. 	
		HU12	Buscador de palabras	<ul style="list-style-type: none"> • Implementar un input para ingreso de palabras junto con un botón de ejecución de búsqueda. • Codificar funcionalidad de búsqueda en el API Rest y en la Interfaz de usuario. • Comprobar que el resultado de la búsqueda sea congruente con los datos registrados en la BD. • Añadir esta funcionalidad en la pantalla principal de visualización de registros. 	
SB04	<i>Pruebas y despliegue a producción</i>	N/A	N/A	<ul style="list-style-type: none"> • Realizar pruebas de aceptación para el <i>dashboard web</i>. • Desplegar el proyecto <i>backend</i> a producción. • Desplegar el proyecto <i>frontend</i> a producción. • Verificar el funcionamiento de los proyectos en conjunto. 	20H
Documentación					30H
TOTAL DE HORAS					240H

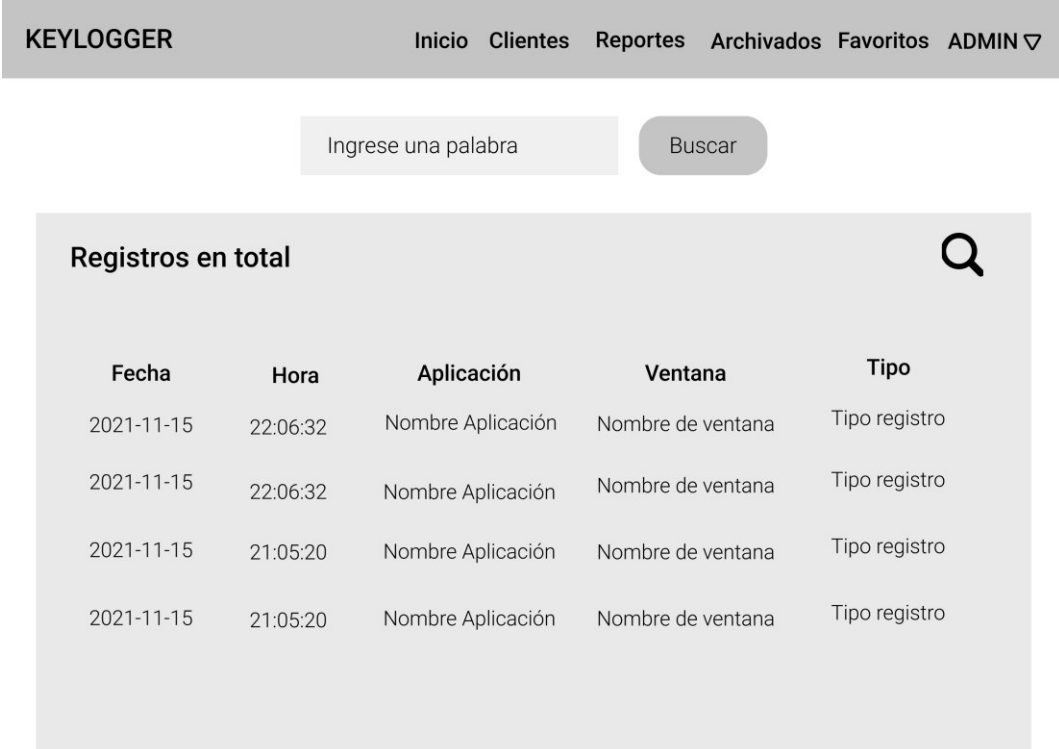
Prototipos de Interfaces

En esta sección se presentan, desde la **Fig. 40** hasta la **Fig. 46**, el resto de los prototipos realizados para el diseño de interfaces del *dashboard* web.



El prototipo muestra un formulario de inicio de sesión con un ícono de candado en la parte superior. El título es "Ingresar al Dashboard". Hay dos campos de entrada: "Correo del administrador" y "Contraseña". Debajo de los campos hay un botón "Ingresar".

Fig. 40 Prototipo de interfaz - Formulario de inicio de sesión



El prototipo muestra la página principal de registros. En la parte superior hay un menú de navegación con "KEYLOGGER" y "Inicio Clientes Reportes Archivados Favoritos ADMIN ▾". Debajo hay un campo de búsqueda con el texto "Ingrese una palabra" y un botón "Buscar". El contenido principal es un encabezado "Registros en total" con un ícono de lupa, seguido de una tabla de registros.

Fecha	Hora	Aplicación	Ventana	Tipo
2021-11-15	22:06:32	Nombre Aplicación	Nombre de ventana	Tipo registro
2021-11-15	22:06:32	Nombre Aplicación	Nombre de ventana	Tipo registro
2021-11-15	21:05:20	Nombre Aplicación	Nombre de ventana	Tipo registro
2021-11-15	21:05:20	Nombre Aplicación	Nombre de ventana	Tipo registro

Fig. 41 Prototipo de interfaz - Página de registros principal



Fig. 42 Prototipo de interfaz - Despliegue de registro de pulsaciones de teclas

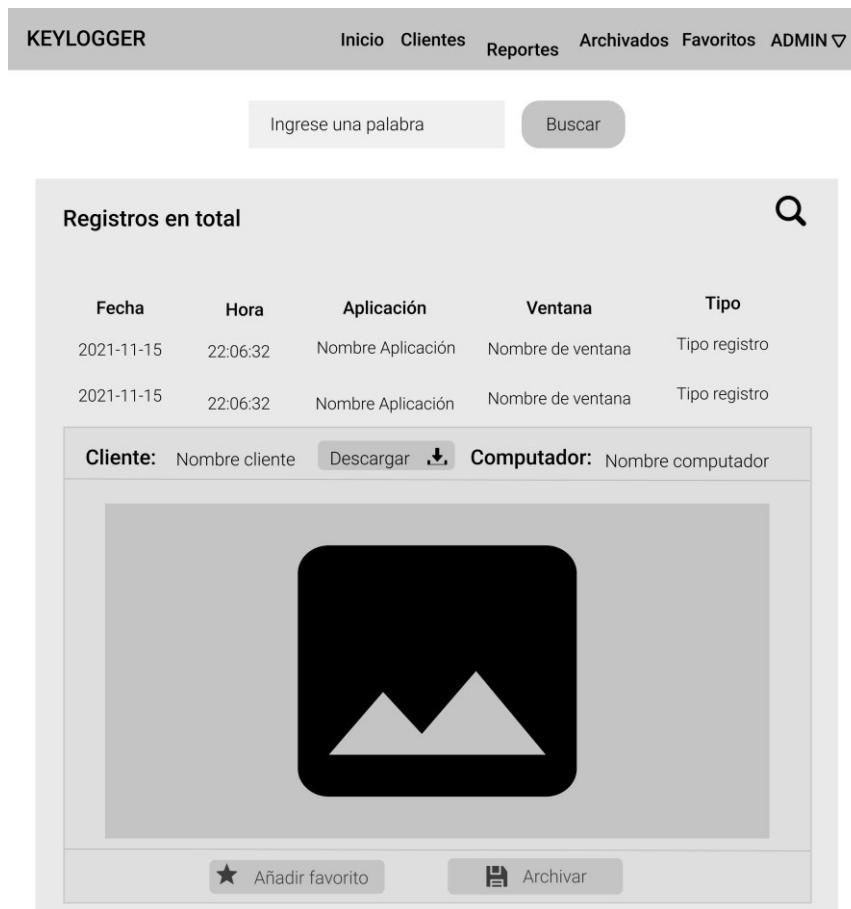


Fig. 43 Prototipo de interfaz - Despliegue de registro de captura de pantalla



Fig. 44 Prototipo de interfaz - Página de registros archivados

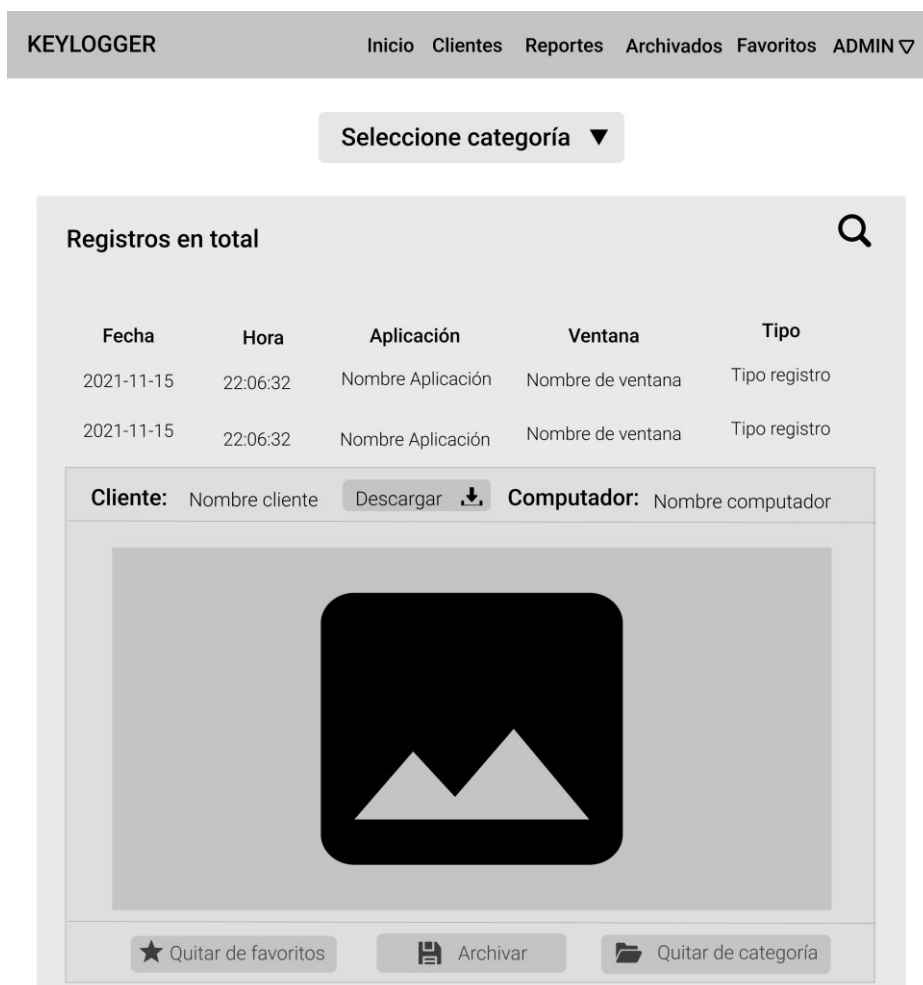


Fig. 45 Prototipo de interfaz - Página de registros favoritos

Clientes en total				Q
Cliente	Computador	Recopilación	Días de actividad	
Nombre de cliente	Nombre Computador	2021-11-15	10	
Nombre de cliente	Nombre Computador	2021-11-15	10	
Nombre de cliente	Nombre Computador	2021-11-15	10	
Nombre de cliente	Nombre Computador	2021-11-15	10	
	Keystrokes	Screenshots	Websites	
# Registros obtenidos:	2	5	6	

Fig. 46 Prototipo de interfaz - Página de clientes

Pruebas de Aceptación

A continuación, se muestran las Pruebas de Aceptación realizadas para verificar el funcionamiento de este proyecto. El propósito que tienen y un ejemplo de estas se ha detallado en el apartado Pruebas de Aceptación. Desde la **TABLA XXV** hasta la **TABLA XLI** se presentan las Pruebas de Aceptación restantes.

TABLA XXV. Prueba de Aceptación Nro. 2 - Visualizar pulsaciones

Prueba de Aceptación	
Identificador: PA02	Identificador Historia de Usuario: HU02
Nombre de Prueba de Aceptación: Visualizar pulsaciones	
Descripción: El administrador del <i>dashboard</i> web puede observar las pulsaciones de teclas que haya realizado el cliente en la tabla de registros principal.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro de tipo <i>keystroke</i> para desplegar su contenido 	
Resultado deseado: El usuario administrador accede a la sección "Registros" por medio del botón del menú principal y se presenta la tabla de registros. Pulsa un registro específico y este se despliega para mostrar su contenido, es decir, las pulsaciones captadas por el <i>keylogger</i> .	
Evaluación de prueba: Resultado exitoso. El contenido del registro se despliega al seleccionarlo y se muestra las pulsaciones de teclas en un cuadro de texto. Aprobación del cliente en un 100%.	

TABLA XXVI. Prueba de Aceptación Nro. 3 - Visualizar capturas de pantalla

Prueba de Aceptación	
Identificador: PA03	Identificador Historia de Usuario: HU03
Nombre de Prueba de Aceptación: Visualizar capturas de pantalla	
Descripción: El administrador del <i>dashboard</i> web puede observar las capturas de pantalla que se hayan obtenido del cliente en la tabla de registros principal.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro de tipo <i>screenshot</i> para desplegar su contenido 	
Resultado deseado: El usuario administrador accede a la sección "Registros" por medio del botón del menú principal y se presenta la tabla de registros. Pulsa un registro específico y este se despliega para mostrar su contenido, es decir, la captura de pantalla obtenida por el <i>keylogger</i> .	
Evaluación de prueba: Resultado exitoso. El contenido del registro se despliega al seleccionarlo y se muestra la captura de pantalla. Aprobación del cliente en un 100%.	

TABLA XXVII. Prueba de Aceptación Nro. 4 - Visualizar pulsaciones de sitios web

Prueba de Aceptación	
Identificador: PA04	Identificador Historia de Usuario: HU04
Nombre de Prueba de Aceptación: Visualizar pulsaciones de sitios web	
Descripción: El administrador del <i>dashboard</i> web puede observar las pulsaciones de teclas que se hayan obtenido, de un sitio o página web visitado por el cliente, en la tabla de registros principal.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro de tipo <i>website</i> para desplegar su contenido 	
Resultado deseado: El usuario administrador accede a la sección "Registros" por medio del botón del menú principal y se presenta la tabla de registros. Pulsa un registro específico y este se despliega para mostrar su contenido, es decir, las pulsaciones de teclas realizadas en un sitio o página web.	
Evaluación de prueba: Resultado exitoso. El contenido del registro se despliega al seleccionarlo y se muestran las pulsaciones de teclas realizadas por el cliente dentro de un sitio o página web. Aprobación del cliente en un 100%.	

TABLA XXVIII. Prueba de Aceptación Nro. 5 - Marcar y desmarcar un registro como favorito

Prueba de Aceptación	
Identificador: PA05	Identificador Historia de Usuario: HU05
Nombre de Prueba de Aceptación: Marcar y desmarcar un registro como favorito	
Descripción: El administrador del <i>dashboard</i> web puede marcar un registro como favorito y enviarlo a la sección de "Favoritos", si así lo desea, puede desmarcarlo como tal y enviarlo de vuelta a la tabla de registros principal.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación - <i>Marcar como favorito:</i> • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro que desee marcar como favorito para desplegar su contenido • Dar clic en el botón "Añadir a favoritos" • Seleccionar la opción "No asignar categoría" • Dar clic en el botón "Añadir" • Acceder a la sección "Favoritos" del <i>dashboard</i> • Seleccionar la opción "Favoritos sin categoría" del selector para mostrar el listado de los registros favoritos - <i>Desmarcar como favorito:</i> • Acceder a la sección "Favoritos" del <i>dashboard</i> • Seleccionar la opción "Favoritos sin categoría" del selector • Pulsar el registro que desee eliminar de favoritos para desplegar su contenido • Dar clic en el botón "Quitar de favoritos" 	

<p>Resultado deseado: -Marcar como favorito: El registro se envía a la sección "Favoritos" del <i>dashboard</i> una vez que el usuario administrador haya pulsado el botón "Añadir" con la opción "No asignar categoría" seleccionada previamente. - Desmarcar como favorito: El registro se presenta nuevamente en la tabla de registros principal después de que el usuario administrador haya pulsado el botón "Quitar de favoritos".</p>
<p>Evaluación de prueba: Resultado exitoso. El registro se presenta en la sección "Favoritos" después de haber sido marcado como tal, es enviado de vuelta a la tabla principal en el momento en que se lo quita de la lista de favoritos. Aprobación del cliente en un 100%.</p>

TABLA XXIX. Prueba de Aceptación Nro. 6 - Visualizar listado de clientes

Prueba de Aceptación	
Identificador: PA06	Identificador Historia de Usuario: HU06
Nombre de Prueba de Aceptación: Visualizar listado de clientes	
Descripción: El administrador del <i>dashboard</i> web puede observar el listado de clientes junto con el número total de registros obtenidos de cada uno.	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Clientes" del <i>dashboard</i> • Pulsar el registro del cliente para desplegar la información referente a los registros obtenidos de este 	
Resultado deseado: El usuario administrador accede a la sección "Clientes" por medio del botón del menú principal y se presenta la tabla de clientes. Pulsa un registro específico y este se despliega para mostrar su contenido, es decir, el número total de registros obtenidos de ese cliente junto con la cantidad específica de estos según el tipo al que pertenecen.	
<p>Evaluación de prueba: Resultado exitoso. El contenido del registro se despliega al seleccionarlo y se muestra el número total de registros obtenidos del cliente además de la cantidad de registros de cada tipo. Aprobación del cliente en un 100%.</p>	

TABLA XXX. Prueba de Aceptación Nro. 7 - Ordenamiento de registros por parámetros

Prueba de Aceptación	
Identificador: PA07	Identificador Historia de Usuario: HU07
Nombre de Prueba de Aceptación: Ordenamiento de registros por parámetros	
Descripción: El administrador del <i>dashboard</i> web puede cambiar el orden de presentación de los registros en la tabla según el parámetro que elija (fecha, hora, aplicación, ventana).	

<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar la cabecera de la columna para cambiar el orden de los registros de acuerdo con ese parámetro.
<p>Resultado deseado: El usuario administrador cambia el orden de los registros presentados en la tabla, en forma ascendente o descendente, de acuerdo con la cabecera de la columna que presione.</p>
<p>Evaluación de prueba: Resultado exitoso. El orden de los registros cambia conforme al parámetro elegido. Aprobación del cliente en un 100%.</p>

TABLA XXXI. Prueba de Aceptación Nro. 8 - Búsqueda de registros por parámetros

Prueba de Aceptación	
Identificador: PA08	Identificador Historia de Usuario: HU08
Nombre de Prueba de Aceptación: Búsqueda de registros por parámetros	
Descripción: El administrador del <i>dashboard</i> web puede localizar el o los registros que coincidan con el término de búsqueda ingresado, la búsqueda se realiza entre los valores de todos los parámetros (fecha, hora, aplicación, ventana).	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Dar clic en el ícono de lupa ubicado en la esquina superior derecha de la tabla • Ingresar el término que se quiere encontrar en el cuadro de texto ubicado en la esquina superior izquierda de la tabla • Pulsar el ícono de equis para ocultar el cuadro de texto y finalizar la búsqueda 	
Resultado deseado: La tabla principal muestra aquellos registros, que, entre los valores de cada parámetro, guarden semejanza con el término ingresado en el cuadro de texto.	
<p>Evaluación de prueba: Resultado exitoso. La tabla muestra solo los registros coincidentes y vuelve a presentar todos los registros al finalizar con la búsqueda. Aprobación del cliente en un 90%.</p>	

TABLA XXXII. Prueba de Aceptación Nro. 9 - Visualizar registros con sus parámetros

Prueba de Aceptación	
Identificador: PA09	Identificador Historia de Usuario: HU09
Nombre de Prueba de Aceptación: Visualizar registros con sus parámetros	
Descripción: El administrador del <i>dashboard</i> web puede visualizar las propiedades de cada registro en la tabla principal. Estas propiedades son referentes a la obtención de cada uno; fecha, hora, nombre de aplicación, título de ventana, tipo de registro, nombre del cliente y nombre del computador.	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> 	

<ul style="list-style-type: none"> • Visualizar el valor de cada parámetro de los registros • Pulsar un registro para desplegar su contenido y parámetros adicionales de este
<p>Resultado deseado: La tabla principal presenta los parámetros mencionados, junto con su valor, de cada registro.</p>
<p>Evaluación de prueba: Resultado exitoso. La tabla muestra todos los registros con sus propiedades definidas en cada columna y dentro del contenido de cada uno. Aprobación del cliente en un 90%.</p>

TABLA XXXIII. Prueba de Aceptación Nro. 10 - Marcar y desmarcar un registro como archivado

Prueba de Aceptación	
Identificador: PA10	Identificador Historia de Usuario: HU10
Nombre de Prueba de Aceptación: Marcar y desmarcar un registro como archivado	
Descripción: El administrador del <i>dashboard</i> web puede marcar un registro como archivado y enviarlo a la sección de "Archivados", si así lo desea, puede desmarcarlo como tal y enviarlo de vuelta a la tabla de registros principal.	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la aplicación - <i>Marcar como archivado:</i> • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro que desee marcar como archivado para desplegar su contenido • Dar clic en el botón "Archivar" • Acceder a la sección "Archivados" del <i>dashboard</i> • Visualizar los registros archivados - <i>Desarchivar registro:</i> • Acceder a la sección "Archivados" del <i>dashboard</i> • Pulsar el registro que desee desarchivar para desplegar su contenido • Dar clic en el botón "Desarchivar registro" 	
<p>Resultado deseado:</p> <ul style="list-style-type: none"> - <i>Marcar como archivado:</i> El registro se envía a la sección "Archivados" del <i>dashboard</i> una vez que el usuario administrador haya pulsado el botón "Archivar". - <i>Desmarcar como favorito:</i> El registro se presenta nuevamente en la tabla de registros principal después de que el usuario administrador haya pulsado el botón "Desarchivar". 	
<p>Evaluación de prueba: Resultado exitoso. El registro se presenta en la sección "Archivados" después de haber sido marcado como tal, es enviado de vuelta a la tabla principal en el momento en que se lo quita de la lista de archivados. Aprobación del cliente en un 100%.</p>	

TABLA XXXIV. Prueba de Aceptación Nro. 11 - Crear categorías

Prueba de Aceptación	
Identificador: PA11	Identificador Historia de Usuario: HU11
Nombre de Prueba de Aceptación: Crear categorías	
Descripción: El administrador del <i>dashboard</i> web puede crear categorías para clasificar registros antes de ser marcados como favoritos.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar un registro de cualquier tipo para desplegar su contenido • Dar clic en el botón "Añadir a favoritos" • Dar clic en el botón "Nueva categoría" • Completar los campos del formulario presentado (nombre y descripción) • Dar clic en el botón "Crear" 	
Resultado deseado: El usuario administrador puede añadir categorías al apartado de "Favoritos".	
Evaluación de prueba: Resultado exitoso. La nueva categoría creada se muestra en la lista de categorías antes de marcar un registro como favorito y en el selector de la sección "Favoritos". Aprobación del cliente en un 90%.	

TABLA XXXV. Prueba de Aceptación Nro. 12 - Búsqueda de palabras

Prueba de Aceptación	
Identificador: PA12	Identificador Historia de Usuario: HU12
Nombre de Prueba de Aceptación: Búsqueda de palabras	
Descripción: El administrador del <i>dashboard</i> web puede buscar palabras dentro del contenido de los registros de pulsaciones de teclas.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Ingresar una palabra de longitud mayor a cuatro caracteres en el cuadro de texto • Dar clic en el botón "Buscar" o presionar la tecla <i>enter</i> • Visualizar los registros coincidentes con el término ingresado en el cuadro de texto 	
Resultado deseado: La tabla principal despliega aquellos registros que en su contenido tengan palabras semejantes al término de búsqueda, caso contrario, se presenta un mensaje que le indique al usuario que no se han encontrado coincidencias.	
Evaluación de prueba: Resultado exitoso. Los registros que coinciden con el término de búsqueda son automáticamente desplegados para que el usuario administrador pueda verificarlo, la tabla vuelve a presentar todos los registros después que el usuario pulsa el botón "Reset". Aprobación del cliente en un 90%.	

TABLA XXXVI. Prueba de Aceptación Nro. 13 - Descargar imágenes

Prueba de Aceptación	
Identificador: PA13	Identificador Historia de Usuario: HU13
Nombre de Prueba de Aceptación: Descargar imágenes	
Descripción: El administrador del <i>dashboard</i> web puede descargar las capturas de pantalla que contienen los registros de este tipo.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro de tipo <i>screenshot</i> para desplegar su contenido • Dar clic en el botón "Descargar" ubicado alado del parámetro nombre del computador 	
Resultado deseado: La captura de pantalla se descarga en la ruta predeterminada de almacenamiento al pulsar el botón "Descargar" del registro desplegado.	
Evaluación de prueba: Resultado exitoso. La captura de pantalla se descarga al pulsar el botón y se almacena localmente. Aceptación del cliente en un 100%.	

TABLA XXXVII. Prueba de Aceptación Nro. 14 - Visualizar número de registros por cliente

Prueba de Aceptación	
Identificador: PA14	Identificador Historia de Usuario: HU14
Nombre de Prueba de Aceptación: Visualizar número de registros por cliente	
Descripción: El <i>dashboard</i> web cuenta con un gráfico estadístico que presenta datos relacionados con el número de registros que se han obtenido de cada cliente.	
Pasos de ejecución: <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder al apartado de "Inicio" del <i>dashboard</i> • Posicionar el puntero del ratón sobre el segmento del cliente en el gráfico • Visualizar la cantidad de registros total del cliente seleccionado 	
Resultado deseado: El gráfico estadístico muestra la proporción de cada cliente de acuerdo con el número de registros obtenidos de cada uno y reacciona al posicionamiento del puntero sobre este, mostrando así el número de registros.	
Evaluación de prueba: Resultado exitoso. El gráfico presenta los datos en tiempo real y muestra la cantidad de registros obtenidos de cada cliente al posicionar el puntero del ratón sobre cada segmento de este. Aceptación del cliente en un 100%.	

TABLA XXXVIII. Prueba de Aceptación Nro. 15 - Visualizar número de registros por cliente y tipo

Prueba de Aceptación	
Identificador: PA15	Identificador Historia de Usuario: HU15
Nombre de Prueba de Aceptación: Visualizar número de registros por cliente y tipo	
Descripción: El <i>dashboard</i> web cuenta con un gráfico estadístico por cada tipo de registro existente, tres en total. Cada uno presenta datos relacionados con el número de registros de ese tipo que se han obtenido de un cliente.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder al apartado de "Inicio" del <i>dashboard</i> • Posicionar el puntero del ratón sobre el segmento del cliente en el gráfico deseado • Visualizar la cantidad de registros de ese tipo del cliente seleccionado 	
Resultado deseado: Cada gráfico estadístico muestra la proporción de un cliente de acuerdo con la cantidad de registros obtenidos de ese tipo y reacciona al posicionamiento del puntero sobre este, mostrando así el número total.	
Evaluación de prueba:	
Resultado exitoso.	
Cada uno de los tres gráficos presenta los datos en tiempo real y muestra la cantidad de registros obtenidos de un tipo por cada cliente al posicionar el puntero del ratón sobre cada segmento de este.	
Aceptación del cliente en un 100%.	

TABLA XXXIX. Prueba de Aceptación Nro. 16 - Agregar registro a una categoría

Prueba de Aceptación	
Identificador: PA16	Identificador Historia de Usuario: HU16
Nombre de Prueba de Aceptación: Agregar registro a una categoría	
Descripción: El administrador del <i>dashboard</i> web puede añadir un registro a una categoría existente y enviarlo a la sección de "Favoritos", si así lo desea, puede quitarlo de dicha categoría y enviarlo de vuelta a la tabla de registros principal.	
Pasos de ejecución:	
<ul style="list-style-type: none"> • Ingresar a la aplicación - <i>Añadir a categoría:</i> • Acceder a la sección "Reportes" del <i>dashboard</i> • Pulsar el registro que desee añadir a una categoría para desplegar su contenido • Dar clic en el botón "Añadir a favoritos" • Seleccionar la categoría deseada de la lista • Dar clic en el botón "Añadir" • Acceder a la sección "Favoritos" del <i>dashboard</i> • Seleccionar el nombre de la categoría a la cual se añadió el registro para mostrarlo - <i>Eliminar registro de categoría:</i> • Acceder a la sección "Favoritos" del <i>dashboard</i> • Seleccionar el nombre de la categoría en el selector • Pulsar el registro que desee eliminar de la categoría para desplegar su contenido • Dar clic en el botón "Quitar de categoría" 	

<p>Resultado deseado: - <i>Añadir a categoría:</i> El registro se añade a la categoría después que el usuario administrador haya pulsado el botón "Añadir" con el nombre de la categoría seleccionado previamente. - <i>Eliminar registro de categoría:</i> El registro se presenta nuevamente en la tabla de registros principal después de que el usuario administrador haya pulsado el botón "Quitar de categoría".</p>
<p>Evaluación de prueba: Resultado exitoso. El registro se presenta cuando la categoría a la cual fue asignado ha sido seleccionada en el selector de la sección "Favoritos", es enviado de vuelta a la tabla principal en el momento en que se lo quita de la categoría. Aprobación del cliente en un 95%.</p>

TABLA XL. Prueba de Aceptación Nro. 17 - Gestionar categorías

Prueba de Aceptación	
Identificador: PA17	Identificador Historia de Usuario: HU17
Nombre de Prueba de Aceptación: Gestionar categorías	
Descripción: La sección "Favoritos" del <i>dashboard</i> cuenta con un selector para que el usuario administrador pueda elegir los registros que desea visualizar, ya sean los marcados solo como favoritos o los que se han añadido a alguna categoría.	
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Ingresar a la aplicación • Acceder al apartado de "Favoritos" del <i>dashboard</i> • Seleccionar una opción del selector de favoritos • Visualizar los registros pertenecientes a la selección previa 	
Resultado deseado: Los registros se presentan en la sección de "Favoritos" de acuerdo con la opción que haya sido elegida en el selector, estos se muestran según la categoría a la cual pertenecen o aquellos que solo han sido marcados como favoritos.	
<p>Evaluación de prueba: Resultado exitoso. Los registros son presentados en la tabla de la sección "Favoritos" de acuerdo con la opción escogida en el selector. Aceptación del cliente en un 100%.</p>	

TABLA XLI. Prueba de Aceptación Nro. 18 - Filtrado de registros según su tipo

Prueba de Aceptación	
Identificador: PA18	Identificador Historia de Usuario: HU18
Nombre de Prueba de Aceptación: Filtrado de registros según su tipo	
Descripción: El usuario administrador cuenta con un menú dentro de la tabla de registros principal para filtrar la presentación de los registros de acuerdo con el tipo que pertenezcan estos (<i>keystroke, screenshot, website</i>).	

Pasos de ejecución:

- Ingresar a la aplicación
- Acceder a la sección "Reportes" del *dashboard*
- Pulsar el ícono de menú que se encuentra alado del ícono de búsqueda en la esquina superior derecha de la tabla
- Dar clic en el campo "Tipo" para desplegar las opciones
- Seleccionar una opción de filtro para los registros
- Visualizar los registros correspondientes al filtro aplicado en la tabla principal

Resultado deseado: La tabla presenta los registros que sean correspondientes con el tipo que haya sido seleccionado en el menú de filtrado, la tabla vuelve a presentar todos los registros si el usuario pulsa el botón "Reset" del menú mencionado.

Evaluación de prueba:

Resultado exitoso.

En la tabla se presentan solo los registros que son del tipo seleccionado en el filtro, la tabla vuelve a presentar todos los registros si el usuario pulsa el botón "Reset" o elimina el filtro aplicado.

Aceptación del cliente en un 90%.

ANEXO III. Manual de Usuario

El manual de usuario del *dashboard* web se muestra en un video que se encuentra en YouTube y se puede acceder a él por medio del siguiente enlace:

ANEXO IV. Manual de Instalación

El código fuente y las instrucciones a seguir para la descarga e instalación del API Rest desarrollada en este proyecto se encuentran en GitHub, se puede acceder a este recurso por medio del siguiente enlace: <https://github.com/Alexix69/api-keylogger>.

De igual forma, el código y las instrucciones respectivas para la descarga e instalación del componente *frontend* del *dashboard* se encuentran en el siguiente enlace: <https://github.com/Alexix69/dashboard-keylogger>.

Las credenciales creadas para la autenticación del usuario administrador en el *dashboard* web son las siguientes:

Correo electrónico: administrador.keylogger@dashboard.com

Contraseña: admin2022