

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE UNA APLICACIÓN WEB PARA OFERTAR HORAS DE PRÁCTICAS PREPROFESIONALES

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE

Guillermo Alexander Rivera Guerrero

guillermo.rivera@epn.edu.ec

DIRECTOR: ING. JUAN PABLO ZALDUMBIDE PROAÑO, MSC.

juan.zaldumbide@epn.edu.ec

CODIRECTOR: ING. LUZ MARINA VINTIMILLA JARAMILLO, MSC.

marina.vintimilla@epn.edu.ec

Quito, octubre 2021

CERTIFICACIÓN

Certificamos que el presente trabajo fue desarrollado por el Sr. Rivera Guerrero Guillermo Alexander como requerimiento parcial a la obtención del título de TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE, bajo nuestra supervisión:

Ing. Juan Zaldumbide, MSc.

DIRECTOR DEL PROYECTO

Ing. Marina Vintimilla, MSc

CODIRECTORA DEL PROYECTO

DECLARACIÓN

Yo Rivera Guerrero Guillermo Alexander con CI: 1721949285 declaro bajo juramento que el trabajo 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.

Sin perjuicio de los derechos reconocidos en el primer párrafo del artículo 144 del Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación – COESC-, soy titular de la obra en mención y otorgo una licencia gratuita, intransferible y no exclusiva de uso con fines académicos a la Escuela Politécnica Nacional.

Entrego toda la información técnica pertinente, en caso de que hubiese una explotación comercial de la obra por parte de la EPN, se negociará los porcentajes de los beneficios conforme lo establece la normativa nacional vigente.

GUILLERMO ALEXANDER RIVERA GUERRERO

DEDICATORIA

A Dios, mis padres, mi hermana, familia en general y amigos a quienes llevo siempre en mis oraciones y muy dentro de mi corazón.

GUILLERMO

AGRADECIMIENTO

Primero agradecer a Dios por permitirme culminar una etapa más en mi vida y nunca dejarme solo.

A mi mamá, que siempre con su cariño y amor supo darme las fuerzas y ánimos necesarios para culminar cada etapa de mi vida universitaria.

A mi papá, que con sus consejos y experiencias de vida me permitió tener un punto de vista objetivo y claro de las metas que debo cumplir.

A mi hermana, que aparte de ser mi confidente y soporte siempre tuvo las palabras y acciones correctas para cada situación, sin duda, sin ella esta meta no la hubiera podido alcanzar.

A mis mejores amigos, Cristina, Jennifer y Stalin que me acompañaron en toda mi vida universitaria y personal, siempre apoyándome con esos grandes momentos que solo ellos pueden darme.

GUILLERMO

ÍNDICE DE CONTENIDOS

1	Introducción	1
1.1	Objetivo general	1
1.2	Objetivos específicos	1
1.3	Alcance	2
2	Metodología	3
2.1	Metodología de Desarrollo	3
2.1.1	Roles	3
2.1.2	Artefactos	4
2.2	Diseño de interfaces	6
2.2.1	Herramienta utilizada para el diseño	7
2.2.2	Aplicación Web	7
2.3	Diseño de la arquitectura	8
2.3.1	Patrón arquitectónico	8
2.4	Herramientas de desarrollo	8
3	Resultados y Discusión	11
3.1	<i>Sprint</i> 0. Configuración del entorno de desarrollo.	11
3.1.1	Diseño de la base de datos relacional.	11
3.1.2	Definición de usuarios.	12
3.1.3	Configuración del entorno de desarrollo para <i>Back-end</i> y <i>Front-end</i>	13
3.2	<i>Sprint</i> 1. Implementación del módulo de Administración de cuentas – Implementación del módulo de administración de perfiles de usuario.	14
3.2.1	Implementación de la interfaz principal para visualización de tutorial y demás componentes informativos para los usuarios	15
3.2.2	Implementación y redacción de página de políticas de uso y privacidad. 15	
3.2.3	Implementación de las interfaces de registro de cuenta estudiantil y cuenta empresarial	16

3.2.4	Validación de los campos de datos solicitados para el registro en la base.	17
3.2.5	Implementación de la interfaz de inicio de sesión para los usuarios y validación de credenciales en los registros de la base de datos.	17
3.2.6	Configuración para caducidad del token dado en el inicio de sesión.	18
3.2.7	Implementación de las interfaces de cuenta estudiantil y empresarial. ..	18
3.2.8	Implementación de componente de registro de curriculum de estudiante.	19
3.2.9	Implementación del componente de actualización de datos de usuario estudiantil.....	19
3.3	<i>Sprint 2.</i> Implementación del módulo de administración de publicaciones – Implementación del módulo de búsqueda – Implementación del módulo de comentarios.....	20
3.3.1	Implementación de la interfaz de publicación de oferta por parte de la cuenta empresarial, validación de campos solicitados y registro en la base.	20
3.3.2	Implementación del componente/módulo de publicación de oferta y el apartado donde se visualiza las publicaciones.	21
3.3.3	Configuración del método <i>post</i> para postulación a publicación por parte de los usuarios con estudiantil y visualización de formulario.....	22
3.3.4	Configuración del método <i>get</i> para obtención de postulaciones de usuarios estudiantes y visualización de componente.	23
3.3.5	Diseño del componente y su configuración para visualización de postulaciones de usuario estudiantil y postulaciones a publicaciones de usuarios empresariales.	23
3.3.6	Diseño de la opción de visualización de perfiles de usuario estudiantil. 24	
3.3.7	Diseño del componente para actualización del estado de postulación... 25	
3.3.8	Diseño del componente para filtrado de publicaciones y ruta en <i>Back-end</i> . 25	
3.3.9	Configuración de visualización de resultados.	26
3.3.10	Implementación de la interfaz tipo foro para visualización y redacción de comentarios.	27

3.3.11	Configuración de los recursos para obtención de usuario creador de comentario.....	27
3.4	<i>Sprint</i> 3. Implementación del módulo del usuario Administrador – Implementación del módulo de contacto con el administrador.....	28
3.4.1	Implementación de la interfaz de control de Administrador.....	28
3.4.2	Validación de los campos para registro de datos de nuevo usuario en la base. 29	
3.4.3	Visualización de comentarios para eliminar y configuración del método <i>delete</i> para elidir dicho registro de la base.....	29
3.4.4	Diseño del contacto con el administrador - desarrollador dentro del componente <i>footer</i> y configuración del correo del desarrollador.	30
3.4.5	Visualización de alerta para envío de correo.	30
3.5	<i>Sprint</i> 4. Implementación de módulos de visualización de perfiles de usuarios y descripción de Ley – Implementación de alertas y detalle de publicación.....	31
3.5.1	Implementación de la interfaz de descripción de la Ley.....	31
3.5.2	Configuración del método <i>get</i> para visualización de los perfiles de usuarios estudiantes.	32
3.5.3	Configuración de <i>plugin</i> para mejor experiencia de usuario con las alertas. 32	
3.5.4	Configuración del método <i>post</i> para visualización detallada de la publicación realizada por parte del usuario con rol empresarial.	32
3.6	<i>Sprint</i> 5. Implementación de pruebas a la Aplicación Web – Despliegue completo de la Aplicación Web.....	33
3.6.1	Pruebas Unitarias.....	34
3.6.2	Pruebas de usuario.....	35
3.6.3	Pruebas de rendimiento.....	35
3.6.4	Despliegue de la <i>API</i> en servidor <i>Heroku</i>	35
3.6.5	Despliegue de la aplicación en el <i>host</i> de <i>Vercel</i>	37
4	Conclusiones y Recomendaciones	39
4.1	Conclusiones	39
4.2	Recomendaciones	40

5	REFERENCIAS BIBLIOGRÁFICAS.....	41
6	ANEXOS.....	i
6.1	Manual Técnico	i
6.2	Manual de Usuario.....	i

ÍNDICE DE FIGURAS

Fig. 1: Prototipo de interfaz de inicio de sesión a la Aplicación Web.	7
Fig. 2: Diseño de la arquitectura Web.	8
Fig. 3: Diseño de la base de datos en Power Designer.	12
Fig. 4: Definición de Roles.....	13
Fig. 5: Estructura del proyecto de parte Front-end.	13
Fig. 6: Estructura del proyecto de parte Back-end.....	14
Fig. 7: Componente de tutoriales para nuevos usuarios.....	15
Fig. 8: Página de políticas de uso y privacidad.....	16
Fig. 9: Interfaz de selección de rol previo a registro de usuario.	17
Fig. 10: Formulario de registro de estudiante.	17
Fig. 11: Formulario de registro de empresa.....	17
Fig. 12: Llenado de datos de usuario empresa en la base de datos.....	17
Fig. 13: Módulo de inicio de sesión de usuario.....	17
Fig. 14: Función de validación y caducidad de token.....	18
Fig. 15: Perfil de usuario estudiante.	18
Fig. 16: Componente de registro de curriculum.	19
Fig. 17: Formulario de nueva publicación.	21
Fig. 18: Formulario de nueva publicación.	22
Fig. 19: Formulario de postulación a oferta.	22
Fig. 20: Componente de visualización de postulaciones.	23
Fig. 21: Componente de visualización de postulaciones.	24
Fig. 22: Interfaz de perfil tipo curriculum.....	24
Fig. 23: Componente de actualización de estado.....	25
Fig. 24: Componente de filtrado de publicaciones.	26
Fig. 25: Publicaciones - resultado de filtrado de publicaciones.....	26
Fig. 26: Interfaz de foro.	27
Fig. 27: Configuración de recursos de entidad Comments.	27
Fig. 28: Interfaz administrativa.....	28
Fig. 29: Componente de registro de nuevos usuarios.....	29
Fig. 30: Componente de eliminación de comentario.....	29
Fig. 31: Componente de registro de nuevos usuarios.....	30
Fig. 32: Alerta de comprobación de envío de correo.	30
Fig. 33: Interfaz referente a la Ley Ecuatoriana.	31
Fig. 34: Función para visualización del curriculum.	32

Fig. 35: Página de plugin SweetAlert.....	32
Fig. 36: Visualización de detalle de oferta.	33
Fig. 37: Prueba 1 - Módulo de inicio de sesión.....	34
Fig. 38: Implementación de JMeter.	35
Fig. 39: Creación y configuración del deploy del <i>API</i>	36
Fig. 40: Despliegue del <i>API</i> en <i>Heroku</i>	36
Fig. 41: Configuración de variables de entorno.	37
Fig. 42: Resultado del deploy de la Aplicación Web en Vercel.	37

ÍNDICE DE TABLAS

TABLA I: Definición de roles.	4
TABLA II: Historia de usuario - Publicación de oferta de prácticas.....	5
TABLA III: Herramientas para el desarrollo de la Aplicación Web.....	9

RESUMEN

Dentro del proceso de culminación de la vida académica de todo estudiante universitario se deben desarrollar actividades comunitarias, como lo son las prácticas pre profesionales, que son un requisito obligatorio previo a la obtención del título universitario, dicho requisito genera un factor de prolongación de tiempo para la graduación del estudiante cuando éste no ha desarrollado dichas actividades a tiempo por factores que involucran la dificultad de encontrar una empresa que esté interesada en aceptar practicantes, es por tal motivo que se ha desarrollado una Aplicación Web que permite ofertar horas de prácticas pre profesionales para estudiantes tanto de universidades públicas como privadas.

Para la elaboración del proyecto se detalla en el presente informe la siguiente estructura: como primer punto se tiene el marco que dio lugar a la problemática para el planteamiento de la Aplicación Web con la finalidad de dar una solución viable. La siguiente sección del documento implica la forma en la que se ha elaborado el proceso de desarrollo partiendo desde el levantamiento de requerimientos, definición de la base de datos y funcionalidades junto con la metodología lo que ha permitido llevar un ciclo continuo de desarrollo, para continuar con el proceso de pruebas realizadas a los componentes y módulos que dan lugar a los objetivos definidos en los *Sprints*, posterior a esto, se muestra el proceso de puesta en producción del proyecto para dar finalizado el informe con las conclusiones y recomendaciones.

PALABRAS CLAVE: Prácticas, *Scrum*, *Sprint*, Estudiantes, Empresas

ABSTRACT

Within the culmination process of the academic life of every university student, community activities must be developed, such as pre-professional practices, which are a mandatory requirement prior to obtaining the university degree, this requirement generates a time extension factor for the graduation of the student when he has not developed such activities on time due to factors that involve the difficulty of finding a company that is interested in accepting interns, it is for this reason that a Web Application has been developed that allows offering hours of pre-professional practices for students of both public and private universities.

For the elaboration of the project, the following structure is detailed in this report: the first point is the framework that gave rise to the problem for the approach of the Web Application to provide a viable solution. The next section of the document involves the way in which the development process has been elaborated, starting from the requirements gathering, definition of the database and functionalities together with the methodology, which has allowed a continuous development cycle, to continue with the testing process of the components and modules that give rise to the objectives defined in the *Sprints*. After this, the process of putting the project into production is shown to finish the report with the conclusions and recommendations.

KEYWORDS: Internships, *Scrum*, *Sprint*, Students, Enterprises

1 INTRODUCCIÓN

El presente informe técnico está centrado en el desarrollo de una Aplicación Web que permite ofertar horas de prácticas pre profesionales; implementado bajo el marco de trabajo ágil *Scrum*. El objetivo propuesto para el desarrollo de la Aplicación Web es que se permite ofertar horas de prácticas pre profesionales a estudiantes de universidades tanto públicas como privadas y para aquellas empresas que están en búsqueda de un practicante que cumpla con el perfil solicitado, la facilidad de realizar la oferta por medio de una publicación dentro de la Aplicación Web.

Para que un estudiante universitario pueda obtener su título es necesario cumplir con una serie de requisitos, siendo uno de ellos, el haber realizado una cierta cantidad de horas relacionadas a la ejecución de prácticas pre profesionales [1].

Actualmente existen plataformas que permiten buscar un empleo como es el caso de “opcionempleo” [2] y “jooble” [3], sin embargo, son plataformas internacionales que no están centradas en un solo país, haciendo que el motor de búsqueda tenga que ser filtrado primero por país y luego por locación, significando un proceso tedioso para el usuario, además de que, en el caso de la plataforma “opcionempleo” se tiene que contratar el servicio para poder realizar una publicación de oferta de empleo por parte de una cuenta empresarial y en el caso de “jooble” no cuenta con esa opción.

Sin embargo, en dichas plataformas se explica la “labor” que va a realizar el postulante, más no se detallan las actividades que va a realizar una vez sea contratado, este punto genera desinterés puesto que en ocasiones los estudiantes son enviados a realizar tareas que no están alineadas con su perfil de egreso y que no fomentan a su desarrollo profesional [4].

En Ecuador se cuenta con sistemas de bolsas de empleo generalizadas mas no una que se centre solo en las actividades de prácticas pre profesionales, por ende, a continuación, se detalla el proceso de desarrollo de la Aplicación Web en cuestión.

1.1 Objetivo general

Desarrollar una Aplicación Web que permita ofertar horas de prácticas preprofesionales para estudiantes universitarios en instituciones tanto públicas como privadas.

1.2 Objetivos específicos

- Determinar los requerimientos de la Aplicación Web.

- Diseñar la arquitectura y los prototipos de las interfaces de usuario de la Aplicación Web.
- Codificar los módulos de la Aplicación Web.
- Verificar los módulos de la Aplicación Web.

1.3 Alcance

Las aplicaciones web representan el continuo avance de las comunicaciones y el acceso a la información, estas hacen uso de un servidor para su funcionamiento a través de un navegador y no es necesario tener que instalarlos en los ordenadores [5], almacenando así la información proporcionada dentro de ellos en nubes de almacenamiento masivo.

Con la Aplicación Web que se ha desarrollado en el presente proyecto, el usuario “estudiante” puede obtener acceso a publicaciones que oferten horas de prácticas pre profesionales por parte de estudiantes y en el caso de cuentas empresariales, la opción de realizar una publicación en búsqueda de practicantes, de tal forma que se asegure su registro y *logueo* de forma segura y así garantizar las siguientes acciones dentro de la Aplicación Web.

La Aplicación Web permite a los usuarios con rol empresarial:

- Registro de oferta de prácticas.
- Registro y actualización de datos del perfil.
- Visualización de postulantes por oferta.
- Registro de comentario.
- Visualización de publicaciones de ofertas de carreras en general.
- Visualización de video tutorial.
- Visualización de perfil de postulante.

La Aplicación Web permite a los usuarios con rol estudiantil:

- Registro de postulación.
- Registro de datos del perfil.
- Registro de comentario.
- Visualización de publicaciones de ofertas de carreras en general.
- Filtrar publicaciones de oferta por categoría de interés.
- Registro de curriculum.
- Visualización de video tutorial.

2 METODOLOGÍA

En los procesos de desarrollo de software se hace uso de metodologías ágiles con el objetivo de proporcionar un trabajo que cumpla con las necesidades y requerimientos del cliente, logrando tener un proceso optimizado, que cumpla con los tiempos establecidos y que se adapte a los requisitos cambiantes durante el desarrollo, aumentando la probabilidad de éxito y disminuyendo los riesgos que involucre un proyecto [6].

Para el desarrollo de la Aplicación Web se requiere el uso de una metodología o marco de trabajo que esté orientada a los cambios existentes dentro del proyecto y que además disminuya el riesgo de desarrollo, por ende, es fundamental el uso de *Scrum*, que está orientado a proyectos que conllevan un periodo breve de tiempo además de que agiliza el uso de los recursos obteniendo un trabajo optimizado, organizado y que agrega valor a los productos entregables.

En la continuación se detalla el uso del marco de trabajo ágil en el proceso de desarrollo de la Aplicación Web.

2.1 Metodología de Desarrollo

El marco de trabajo ágil *Scrum* tiene la finalidad de obtener el mejor de los resultados en un proyecto de desarrollo de software, siendo un conjunto de buenas prácticas que permite la definición de roles para especificar la vía por la cual se debe ejecutar un proyecto y así obtener el mejor resultado [7]. Este marco de trabajo propone realizar el desarrollo por medio de ciclos iterativos llamados *Sprints*, los cuales conllevan un tiempo entre una a cuatro semanas para alcanzar las denominadas metas del *Sprint*, significando estos, productos entregables dentro de las tareas previamente establecidas como nuevas versiones del producto que ya pueden ser utilizadas [8], además este marco de trabajo funciona de tal manera que aunque se vayan realizando cambios en los requerimientos durante el desarrollo se mantenga siempre su objetivo apuntando hacia un producto de calidad.

2.1.1 Roles

Dentro de las metodologías ágiles, *Scrum* es una de las más representativas y utilizadas por lo que es totalmente útil en proyectos complejos o que contienen una cantidad definida de participantes [9], por ende, para lograr el éxito y minorizar el riesgo en el

desarrollo del proyecto es necesaria la definición de los roles que se detallan a continuación:

Scrum Master

Es el encargado de liderar el equipo de desarrollo, guiándolos hacia los objetivos propuestos para el proyecto, manteniendo una gestión mínima del riesgo y trabajando de la mano del *Product Owner*.

Product Owner

Representa al cliente final, es aquel que propone las necesidades y los requerimientos que la Aplicación Web va a tener, además de asegurarse que el *Producto Backlog* sea implementado de manera correcta.

Development Team

Es el equipo encargado del desarrollo del producto, por ello, está conformado por personas con conocimientos técnicos para llevar a cabo la elaboración de la Aplicación Web, tomando como punto de partida las historias de usuario previamente descritas por el *Product Owner*.

Los roles definidos con el objetivo de alcanzar el éxito del proyecto se encuentran detallados en la **TABLA I**.

TABLA I: Definición de roles.

Rol	Encargado
<i>Scrum Master</i>	Juan Zaldumbide
<i>Product Owner</i>	Empresas Interesadas en ofertar y Estudiantes Universitarios.
<i>Development Team</i>	Guillermo Rivera

2.1.2 Artefactos

En *Scrum*, los artefactos son desarrollados con la finalidad de dar una transparencia en lo que a información respecta, para que cada miembro del equipo pueda tener entendimiento del artefacto que se va a utilizar logrando realizar una toma correcta de las decisiones [10]. Por consiguiente y partiendo de esta definición, se elabora la Aplicación Web haciendo uso de los siguientes artefactos:

Recopilación de Requerimientos

La recopilación de requerimientos es el punto de partida de todo proyecto de software, es en este punto dónde se analizan las funcionalidades del producto una vez sea lanzado a producción, tomando los requerimientos por parte del cliente y añadiendo las necesidades que presenta.

Para cumplir con esta etapa del desarrollo se establecen encuestas a posibles usuarios finales definiendo un banco de preguntas desarrolladas con la intención de conocer los requerimientos y necesidades que presentan los posibles usuarios finales de la Aplicación Web, dicho proceso se detalla en la sección del Manual Técnico – Recopilación de requerimientos página 2.

Historias de Usuario

Son aquellas que describen las características de un requerimiento por parte de la persona que la solicita. Estas historias de usuario por lo general se escriben en fichas o en notas adhesivas para su posterior almacenaje y organización [11], presentando dentro de ellas cierta información que permite clasificarla en orden de importancia para luego poder discutir sobre la misma.

A continuación, se presenta en la **TABLA II** el modelo de historia de usuario utilizado para la organización y desarrollo del proyecto, todas las historias de usuario se encuentran detalladas en la sección del Manual Técnico – Historias de usuario página 6.

TABLA II: Historia de usuario - Publicación de oferta de prácticas.

Historia de Usuario	
Identificador: HU-01	Usuario: Empresa
Nombre de historia: Publicar oferta de prácticas.	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Guillermo Rivera	
Descripción: Como usuario con cuenta empresarial necesito realizar una publicación en la cual solicito un practicante para ciertas actividades.	
Validación: El usuario puede realizar una publicación, la cual puede editarla y eliminarla en el caso de que sea necesario.	

Product Backlog

Es el conjunto de requerimientos apilados por orden de importancia, además de que fluctúa a lo largo del proyecto debido a los requerimientos cambiantes, dentro de estas listas se encuentran aquellas iteraciones que agregan valor al producto siendo entregables para el cliente.

Dentro de esta lista se encuentran los objetivos y requisitos de la Aplicación Web que son de un nivel alto y que forman parte de las historias de usuario, incluyendo aquellas que involucran un riesgo durante el desarrollo. Es importante resaltar que el cliente al momento de definir el *Product backlog* no cuenta con la obligación de completar en su totalidad los requerimientos, debido a que estos pueden ir surgiendo a lo largo del desarrollo de la Aplicación Web y/o cuando se vayan realizando los entregables, sin embargo, aquellos que ya estén definidos y se encuentren dentro del conjunto de requerimientos tienen un grado de detalle y son los requerimientos principales del producto [12].

Dentro de la elaboración de la Aplicación Web se define el *Product Backlog* a partir de los resultados del levantamiento de requerimientos por medio de la encuesta, dichos requerimientos se encuentran detallados dentro de la sección del Manual Técnico – *Product Backlog* página 15.

Sprint Backlog

Es la agregación del objetivo al *Sprint*, aquellos elementos incorporados en la lista son seleccionados, de tal forma que se detalle un plan para poder darle un incremento al producto entregable como meta del *Sprint*, por lo general este artefacto debe representar actividades que el equipo debe realizar juntando así un conjunto de construcciones o también denominados *ítems* que son entregados logrando el objetivo del *Sprint* en cuestión [13].

La implementación del artefacto dentro del presente proyecto se encuentra detallado en la sección del Manual Técnico – *Sprint Backlog* página 17.

2.2 Diseño de interfaces

El diseño de las interfaces son montajes que permiten a los desarrolladores web implementar un prototipo de la interfaz de usuario concerniente una aplicación o un proyecto de software, de tal manera que permite mostrar al cliente un prototipo de la funcionalidad y la visualización sin la necesidad de mostrarlo de manera física o implementada [14].

Para el diseño del *Front-end* de la Aplicación Web se especifican un conjunto de *mockups* junto con su funcionalidad, las cuales se encuentran detalladas en la sección del Manual Técnico – Diseño de interfaces página 22.

2.2.1 Herramienta utilizada para el diseño

Balsamiq es una herramienta de prototipado de interfaces de usuario, que al ser multiplataforma permite ser utilizada en cualquier sistema operativo, dicho ambiente está diseñado para ayudar al diseño de interfaces, dando como ejemplos, una serie de objetos prediseñados listos para ser utilizados dentro de los prototipos [15].

El conjunto de interfaces referentes al desarrollo de la Aplicación Web, son elaboradas en *Balsamiq* permitiendo entregar en contexto el funcionamiento de la Aplicación Web al cliente, a continuación, se muestran el detalle de la implementación de esta herramienta para la elaboración de la etapa inicial del proyecto.

2.2.2 Aplicación Web

La **Fig. 1** muestra el prototipo de la interfaz de inicio de sesión general de los diferentes usuarios en la Aplicación Web, los siguientes prototipos de interfaces se encuentran detalladas en la sección del Manual Técnico – Diseño de interfaces página 22.

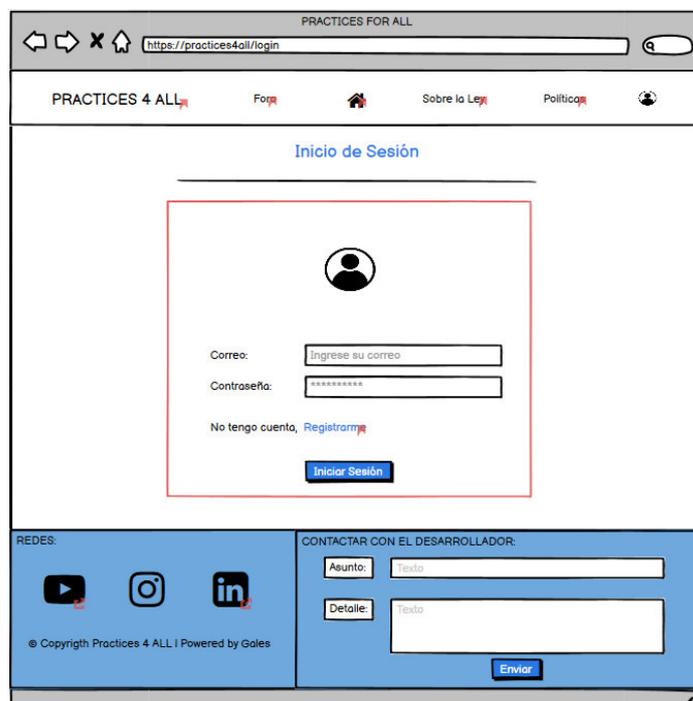


Fig. 1: Prototipo de interfaz de inicio de sesión a la Aplicación Web.

2.3 Diseño de la arquitectura

El diseño de la arquitectura corresponde al boceto de alto nivel de un sistema, el cual permite definir objetivos y actividades a desarrollar por cada módulo que especifique el desarrollo y funcionalidad de un software, especificando un flujo, características y detalles de las diferentes herramientas utilizadas para su elaboración, cabe recalcar que también involucra características como la ubicación de cada componente y la representación abstracta de los diseños tanto de la base de datos como de las interfaces de usuario definidas en la recopilación de requerimientos [16].

2.3.1 Patrón arquitectónico

Un patrón arquitectónico define un esquema de manera estructural para cualquier sistema de software, que permite construir un conjunto de reglas y subsistemas para organizar las relaciones entre estos [17].

En la **Fig. 2** se muestra el patrón arquitectónico llevado a cabo para el desarrollo de la Aplicación Web, detallando las herramientas utilizadas y el flujo de construcción, logrando un control de cada módulo.

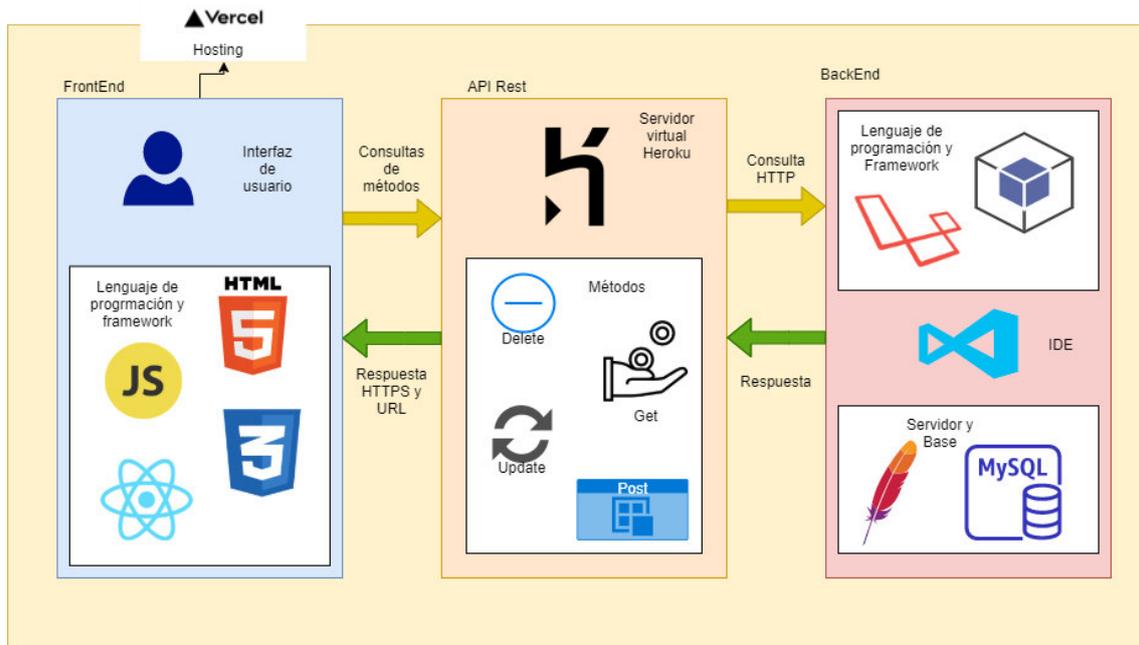


Fig. 2: Diseño de la arquitectura Web.

2.4 Herramientas de desarrollo

Dentro del desarrollo de la Aplicación Web a partir de los requerimientos que se ha levantado y el diseño del prototipo de las interfaces de usuario, en la **TABLA III** se detalla

el conjunto de herramientas que se han utilizado en el desarrollo de la Aplicación Web y su base de datos respectivamente, además, la justificación del porqué del uso de la herramienta.

TABLA III: Herramientas para el desarrollo de la Aplicación Web.

Herramienta	Justificación
Node Js	El uso de esta herramienta facilita la ejecución de servidores debido a su arquitectura basada en eventos que permite la creación de aplicaciones que tipo <i>network</i> que no afecte el rendimiento del servidor [18].
Visual Studio Code	La implementación de este editor de código fuente facilita un entorno de desarrollo con varios lenguajes de programación, incluyendo el manejo de los ficheros y extensiones de los proyectos en los cuales se trabaja [19].
Xampp	El uso de esta herramienta permite verificar el funcionamiento de un proyecto donde no es indispensable una conexión a internet [20], por lo cual permite gestionar los puertos y servidores de apache donde se almacena la base de datos MySQL.
Postman	El uso de esta herramienta posibilita realizar consultas por medio de métodos hacia el <i>API Rest</i> facilitando el monitoreo y realización de pruebas dentro de esta [21].
Heroku	La implementación de <i>heroku</i> provee el <i>upload</i> de la <i>API Rest</i> hacia un servidor privado con el protocolo <i>Https</i> .
Material UI	La implementación de esta normativa permite el uso de librerías tipo CSS para

	el manejo de la interfaz de usuario de la Aplicación Web.
Vercel	El uso de <i>vercel</i> como plataforma en la nube, proporciona un alojamiento para la Aplicación Web de manera instantánea y escalable [22].
React	Es una librería que se ha desarrollado para el lenguaje JavaScript que facilita el desarrollo e implementación de interfaces de usuario, también implementa lo que es el modelo-vista-controlador [23].

3 RESULTADOS Y DISCUSIÓN

Seguidamente, se muestran los resultados pertenecientes a la organización del desarrollo de cada una de las actividades que se han definido para cada *Sprint*, incluyendo la configuración de las herramientas para el desarrollo de la Aplicación Web, abarcando las pruebas de usuario, pruebas unitarias y pruebas de rendimiento, además del despliegue del *API* y la Aplicación Web en los diferentes servidores.

3.1 *Sprint* 0. Configuración del entorno de desarrollo.

Partiendo de la planificación que se realiza en el *Sprint Backlog*, las actividades que involucran al *Sprint* 0 incluyen lo siguiente:

- Diseñar la base de datos relacional.
- Definición de usuarios.
- Configurar el entorno de desarrollo del *Back-end* y *Front-end*.

3.1.1 Diseño de la base de datos relacional.

El diseño de la base de datos implementada a través de la herramienta *Power Designer*, cuya interfaz permite tener un acercamiento directo a los modelos entidad-relación que se desarrollen, dentro del diseño se han definido 6 entidades, las cuales son:

- Usuarios
- Publicaciones
- Comentarios
- Categorías
- Postulaciones
- Curriculums

Sin embargo, en la entidad *users* se manejan relaciones polimórficas por lo cual se generan dos tablas que en sí no muestran una relación directa en la base con las otras entidades pero que, si almacenan información de los usuarios con rol empresa y administrador, a parte de esos roles se maneja uno tercero, el rol estudiante, el cual almacena sus datos en la entidad principal *Users*, las tablas de relación polimórfica son:

- *Admins*
- *Businesses*

En la **Fig. 3** se observa el modelo físico de la base de datos que se ha implementado con una totalidad de 8 entidades, más las creadas por las relaciones entre las tablas y

las diferentes entidades que se han creado automáticamente al momento de inicializar el proyecto en *Laravel*.

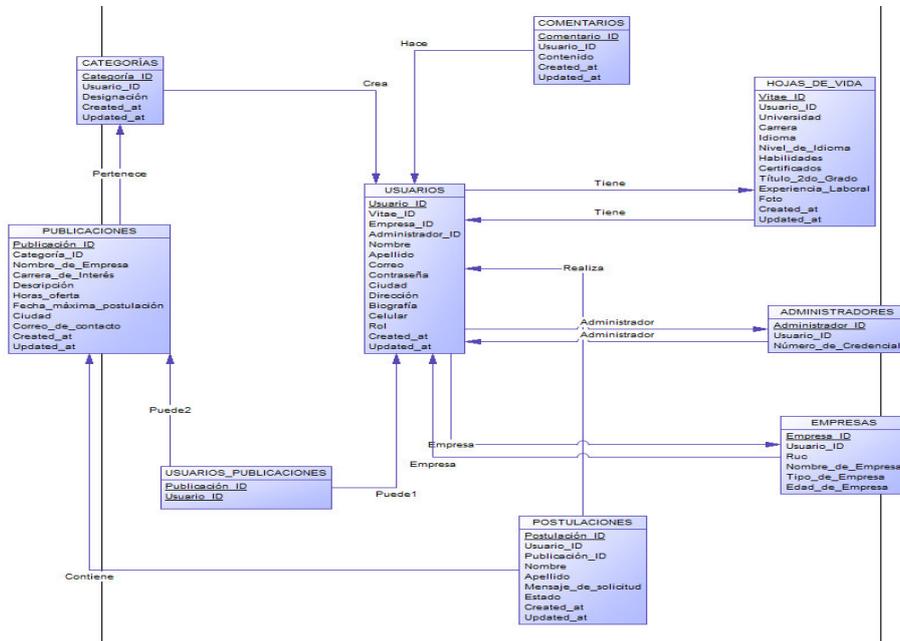


Fig. 3: Diseño de la base de datos en *Power Designer*.

3.1.2 Definición de usuarios.

La Aplicación Web maneja diferentes usuarios, por lo cual en la **Fig. 4** se ilustra dichos roles junto con las actividades principales que pueden realizar dentro de los diferentes componentes y módulos que se implementan en la interfaz de usuario, esto cuando el usuario en cuestión tiene una cuenta registrada y se encuentra con sesión activa.

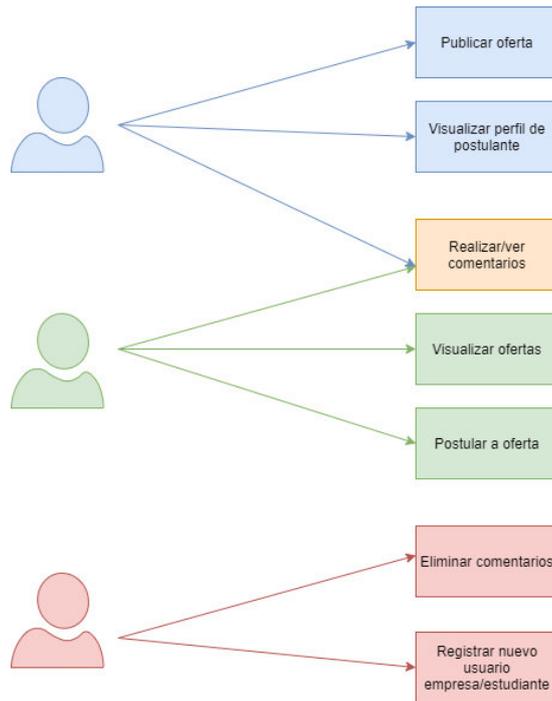


Fig. 4: Definición de Roles.

3.1.3 Configuración del entorno de desarrollo para *Back-end* y *Front-end*.

Visual Studio Code es el *IDE* implementado para el desarrollo de ambas partes del proyecto tanto *Back-end* como *Front-end*, sin embargo, la estructura del proyecto es diferente para cada caso por lo cual se ha utilizado extensiones específicas por lenguaje de programación dentro del desarrollo de la Aplicación Web.

La **Fig. 5** y la **Fig. 6** muestran la estructura de los proyectos.

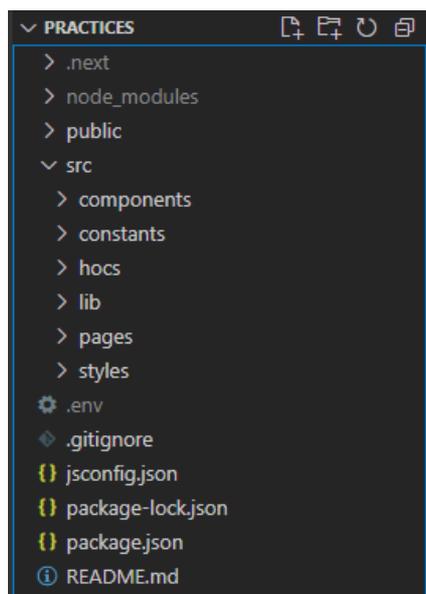


Fig. 5: Estructura del proyecto de parte *Front-end*.

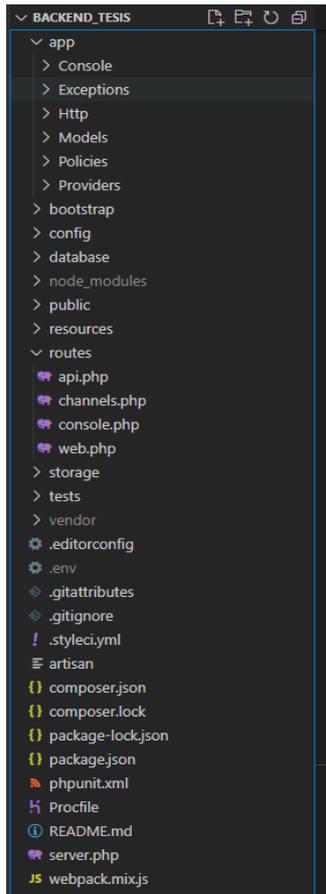


Fig. 6: Estructura del proyecto de parte *Back-end*.

Tomando en cuenta que se hace uso de levantamiento de servidores para poder correr el localhost y probar el desarrollo, se ha configurado *Xampp* para levantar el servidor de apache en el puerto 80 y poder acceder a la base de datos MySQL.

3.2 *Sprint* 1. Implementación del módulo de Administración de cuentas – Implementación del módulo de administración de perfiles de usuario.

Siguiendo la planificación que se establece en el *Sprint backlog* que se detalla en Manual Técnico, a continuación, se especifican las actividades a realizar pertenecientes al *Sprint* 1:

- Implementación de la interfaz principal para visualización de tutorial y demás componentes informativos para los usuarios.
- Implementación y redacción de la página de políticas de uso y privacidad.
- Implementación de las interfaces de registro de cuenta estudiantil, cuenta empresarial y registro de datos en la base.
- Validación de los datos solicitados para el ingreso en la base.

- Implementación de la interfaz de inicio de sesión de los usuarios y validación de credenciales en los registros de la base de datos.
- Configuración para caducidad del token dado al inicio de sesión.
- Implementación de las interfaces de perfil de cuenta estudiantil y empresarial.
- Implementación de componente de registro de curriculum de estudiante.
- Implementación del componente de actualización de datos de usuario estudiante.

3.2.1 Implementación de la interfaz principal para visualización de tutorial y demás componentes informativos para los usuarios.

El manejo de roles dentro de la Aplicación Web puede causar cierta confusión tratándose de un correcto uso, es por tal razón que, la interfaz principal contiene dos videos tutoriales que interpretan un conjunto de pasos que, dependiendo del rol, el usuario que ha iniciado sesión debe seguir para poder hacer un correcto uso de las características y funcionalidades de la Aplicación Web tal como lo indica la **Fig. 7**, además contiene un conjunto de componentes informativos que tienen la intención de llamar la atención de los usuarios para que hagan uso de la Aplicación Web, encontrándose distribuidos a lo largo de la interfaz principal.

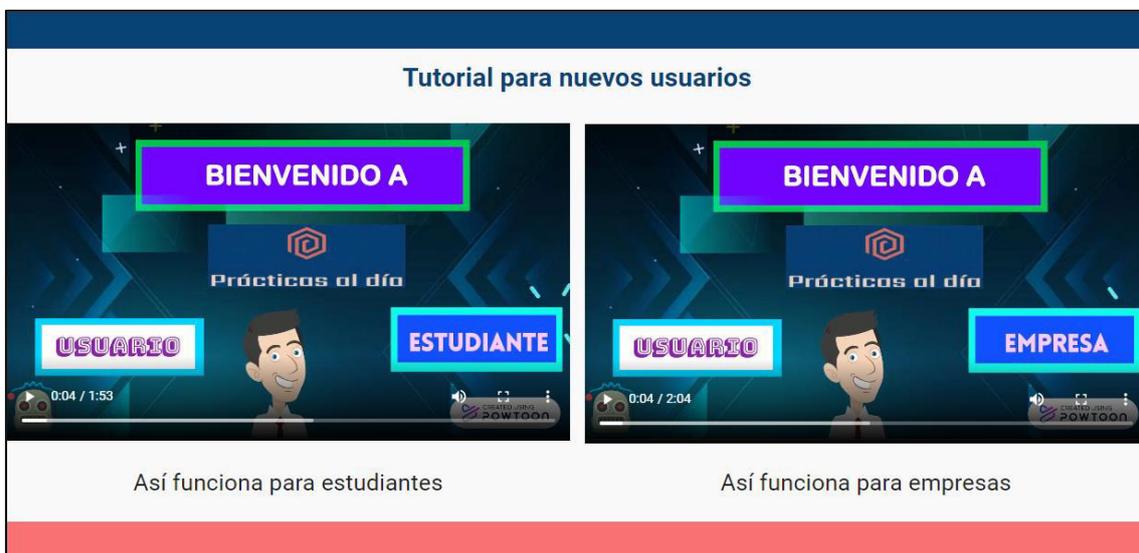


Fig. 7: Componente de tutoriales para nuevos usuarios.

3.2.2 Implementación y redacción de página de políticas de uso y privacidad.

La Aplicación Web incorpora una página de acceso público que le permite al usuario conocer cómo va a ser el manejo de los datos una vez sean ingresados, esta

implementación es realizada con la intención de ser una Aplicación Web lo más transparente posible con los usuarios, la **Fig. 8** ilustra la página mencionada.



Fig. 8: Página de políticas de uso y privacidad.

3.2.3 Implementación de las interfaces de registro de cuenta estudiantil y cuenta empresarial.

Debido al manejo de roles de usuarios en la aplicación, se ha definido dos tipos de registros con cierta diferencia en los campos solicitados, previo al registro se indica al usuario una interfaz de registro general que le permite la redirección al formulario de registro dependiendo del rol que vaya a cumplir tal como se muestra en la **Fig. 9**.



Fig. 9: Interfaz de selección de rol previo a registro de usuario.

La **Fig. 10** y la **Fig. 11** muestran los formularios de registro de los roles estudiante y empresa.

Fig. 10: Formulario de registro de estudiante.

Fig. 11: Formulario de registro de empresa.

3.2.4 Validación de los campos de datos solicitados para el registro en la base.

Los campos que se muestran en los formularios de registro cuentan con una validación que permite que se llenen todos los datos por parte del usuario previo al registro, de tal manera que se mantenga una integridad dentro de la entidad *users* en la base de datos.

La **Fig. 12** muestra cómo se almacenan los datos de empresa que se solicitan en el formulario de registro de este rol.



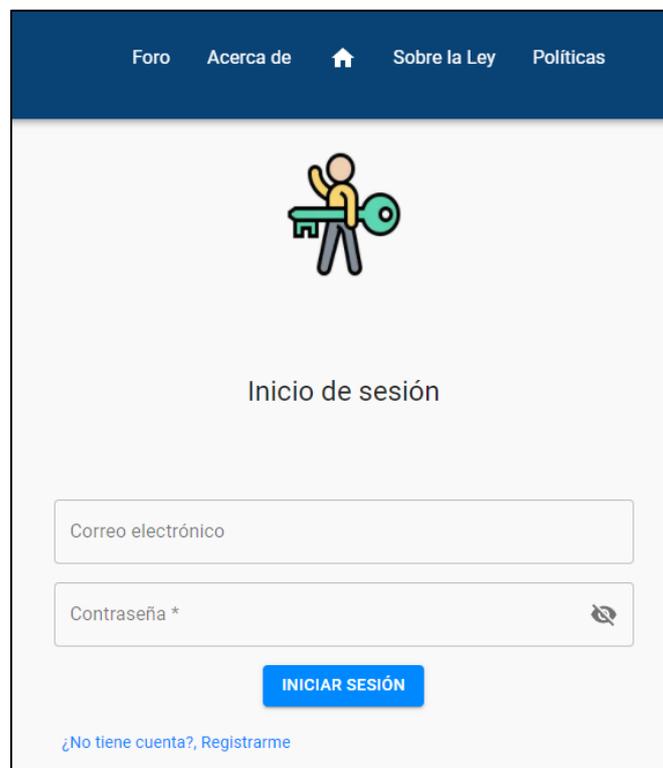
The screenshot shows a web interface for a data table. At the top, there are controls for 'Mostrar todo', 'Número de filas: 25', 'Filtrar filas: Buscar en esta tabla', and 'Sort by key: Ninguna'. Below this is a '+ Opciones' link and a table with columns: 'id', 'ruc', 'business_name', 'business_type', and 'business_age'. The table contains two rows of data. Below the table, there are actions: 'Seleccionar todo', 'Para los elementos que están marcados: Editar, Copiar, Borrar, Exportar'. At the bottom, there are the same controls as at the top.

	id	ruc	business_name	business_type	business_age
<input type="checkbox"/>	1	1234567890123	Martina Tech	Tecnológico	30 años
<input type="checkbox"/>	2	1234567890123	Dicomsa	Tecnología	15 años

Fig. 12: Llenado de datos de usuario empresa en la base de datos.

3.2.5 Implementación de la interfaz de inicio de sesión para los usuarios y validación de credenciales en los registros de la base de datos.

La **Fig. 13** ilustra la interfaz de inicio de sesión, dicha interfaz es general sin depender del rol del usuario debido a que la Aplicación Web realiza una validación de existencia del correo y contraseña del usuario en la base de datos, de tal manera que, si existe el registro, se devuelve un token al usuario para que pueda tener acceso autorizado a la Aplicación Web.



The screenshot shows a user login interface. At the top, there is a dark blue navigation bar with links: 'Foro', 'Acerca de', a home icon, 'Sobre la Ley', and 'Políticas'. Below the navigation bar is a large white area with a central illustration of a person holding a key. Underneath the illustration is the text 'Inicio de sesión'. Below this are two input fields: 'Correo electrónico' and 'Contraseña *'. The 'Contraseña *' field has a toggle icon for visibility. Below the input fields is a blue button labeled 'INICIAR SESIÓN'. At the bottom, there is a link: '¿No tiene cuenta?, Regístrame'.

Fig. 13: Módulo de inicio de sesión de usuario.

3.2.6 Configuración para caducidad del token dado en el inicio de sesión.

La Fig. 14 muestra la función a nivel de *Front-end* que permite la validación y caducidad del token del usuario que realiza la petición *post* a la ruta de inicio de sesión definida en el *Back-end* de la Aplicación Web.

```
const handleUser = (user) => {
  if (user) {
    // si tengo sesión activa
    setUser(user);
    cookie.set("auth", true, {
      expires: 1, // duración de un día con actividad
    });
  }
};
```

Fig. 14: Función de validación y caducidad de token.

3.2.7 Implementación de las interfaces de cuenta estudiantil y empresarial.

La Fig. 15 indica la interfaz de perfil de la cuenta estudiantil, debido al manejo de roles que se maneja en la Aplicación Web, se implementa una validación a nivel de autenticación que permite obtener el rol del usuario con sesión activa, de tal manera que redirige a dicho usuario a la interfaz de perfil que corresponda.

Perfil de Usuario

Información:

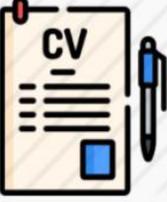


Nombre: Guillermo
Apellido: Rivera
Correo electrónico: guillermo@rivera.com
Celular: 0960625886
Provincia: Pichincha
Ciudad: Quito

Biografía: Soy estudiante de Desarrollo de Software en la EPN, me gusta la música electrónica y las rolas suaves además de ver animé

[VER OFERTAS](#) [ACTUALIZAR DATOS](#)

Curriculum



No hay registro

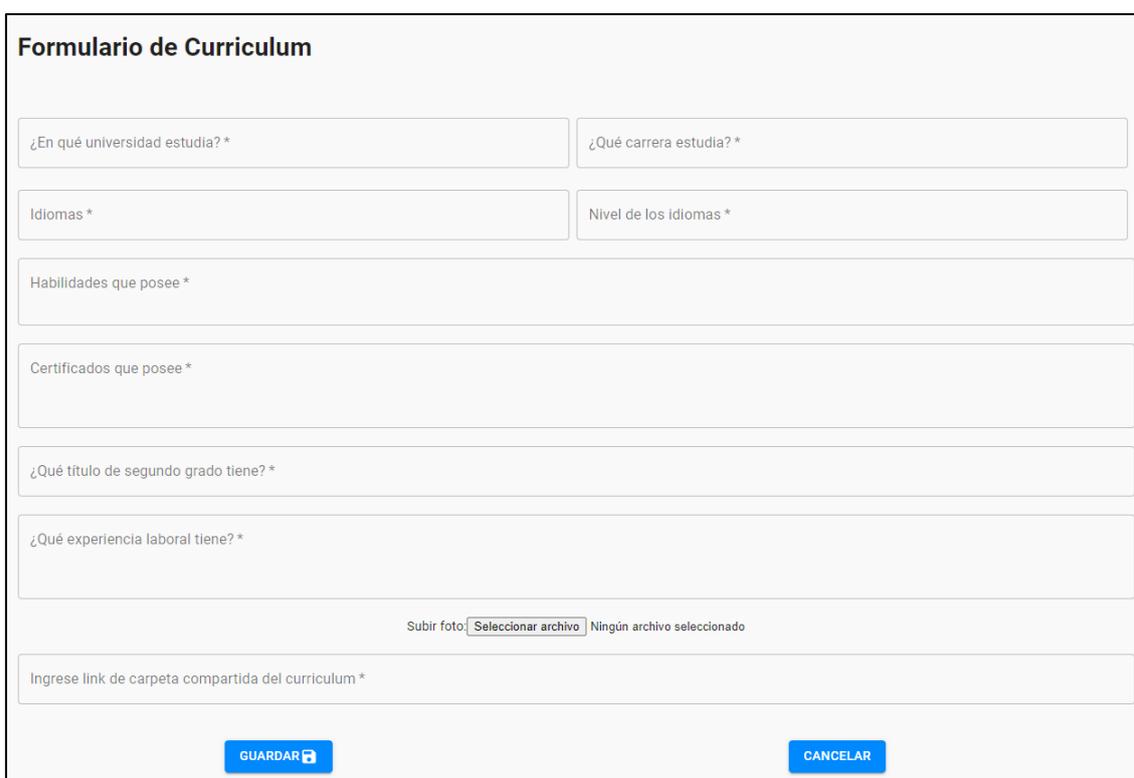
Recuerde, es recomendable postular con un curriculum lleno, de tal manera que la empresa puede revisarlo.

[LLENAR UNO](#)

Fig. 15: Perfil de usuario estudiante.

3.2.8 Implementación de componente de registro de curriculum de estudiante.

El usuario con rol estudiantil posterior a su registro en la Aplicación Web y previo a la postulación a una oferta, puede registrar sus datos profesionales por medio de un componente como lo indica la **Fig. 16**, de esta manera se permite el ingreso de su curriculum para que sea visualizado tanto en su interfaz de perfil como por el usuario con rol empresarial que ha realizado la publicación de la oferta en la que el usuario estudiante desea postular.



Formulario de Curriculum

¿En qué universidad estudia? *

¿Qué carrera estudia? *

Idiomas *

Nivel de los idiomas *

Habilidades que posee *

Certificados que posee *

¿Qué título de segundo grado tiene? *

¿Qué experiencia laboral tiene? *

Subir foto: Ningún archivo seleccionado

Ingrese link de carpeta compartida del curriculum *

Fig. 16: Componente de registro de curriculum.

3.2.9 Implementación del componente de actualización de datos de usuario estudiante.

Dentro del perfil del usuario con rol estudiantil se presenta el módulo de actualización de datos, el cual, muestra la información actual almacenada en la base, sin embargo, se permite solamente la actualización de ciertos campos, incluyendo aquellos que se han considerado pertinentes acerca del curriculum una vez este haya sido registrado, por ende, los componentes se encuentran validados dependiendo del tipo de actualización que se requiera por parte del usuario, con un re direccionamiento una vez haya sido concluida la acción en cuestión.

3.3 *Sprint* 2. Implementación del módulo de administración de publicaciones – Implementación del módulo de búsqueda – Implementación del módulo de comentarios.

Siguiendo la planificación establecida en el *Sprint backlog* que se detalla en el Manual Técnico, a continuación, se muestra las actividades concernientes al *Sprint* 2.

- Implementación del componente/módulo de publicación de oferta por parte de la cuenta empresarial, validación de campos solicitados y registro en la base.
- Implementación de la interfaz y del apartado donde se visualiza las publicaciones.
- Configuración del método *post* para postulación a publicación por parte de los usuarios con rol estudiantil y visualización de formulario.
- Configuración de método *get* para obtención de postulaciones de usuarios estudiantes y visualización de componente.
- Diseño del componente y su configuración para visualización de postulaciones de usuario estudiante y postulaciones a publicaciones de usuarios empresariales.
- Implementación de la opción de visualización de perfiles de usuario estudiante.
- Diseño del componente de actualización de estado de postulación.
- Diseño del componente para filtrado de publicaciones.
- Configuración de visualización de resultados.
- Implementación de la interfaz tipo foro de visualización y realización de comentario.
- Configuración del método *delete* para eliminación de comentario en la base de datos.

3.3.1 Implementación de la interfaz de publicación de oferta por parte de la cuenta empresarial, validación de campos solicitados y registro en la base.

El usuario con rol empresarial puede ofertar horas de prácticas pre profesionales, por lo cual debe llenar un formulario con los datos que se solicita para el almacenaje en la base de datos tal como se muestra en la **Fig. 17**, la finalidad de los campos del componente se detalla en el Manual de Usuario.

Nueva Publicación de oferta

💡 ¿CÓMO LLENAR?

📅

▼

PUBLICARCANCELAR

Fig. 17: Formulario de nueva publicación.

3.3.2 Implementación del componente/módulo de publicación de oferta y el apartado donde se visualiza las publicaciones.

El usuario con rol estudiantil a través el menú que se muestra en el avatar del componente de navegación accede a su perfil el cual contiene la funcionalidad principal de la aplicación, donde se encuentra un botón que redirecciona a la interfaz donde se encuentran las publicaciones de ofertas realizadas por los usuarios con rol empresarial, la **Fig. 18** muestra la interfaz y los componentes incorporados para mejorar la interacción del usuario con la Aplicación Web.

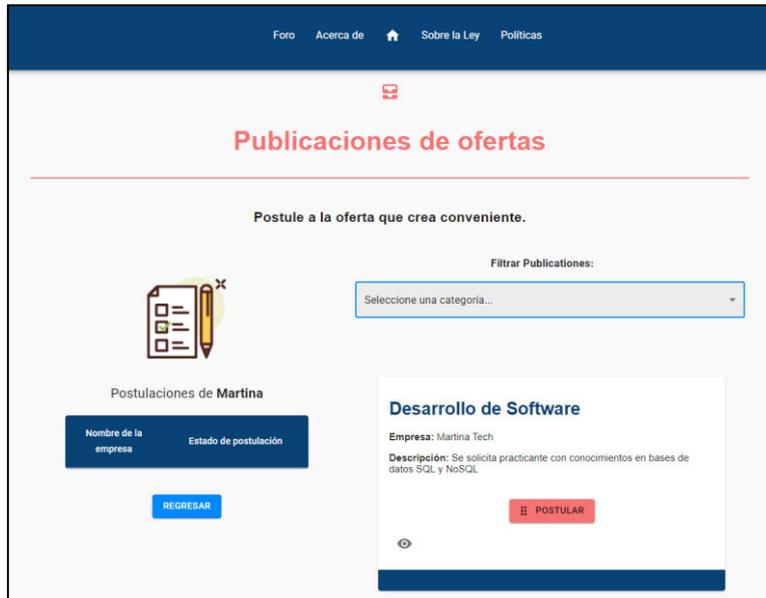


Fig. 18: Formulario de nueva publicación.

3.3.3 Configuración del método *post* para postulación a publicación por parte de los usuarios con estudiantil y visualización de formulario.

La Fig. 19 detalla el formulario de postulación a una oferta, dicho formulario capta el identificador de la postulación que el usuario ha seleccionado, de tal forma que permita solo llenar los campos solicitados a manera de petición a dicha oferta de publicación, la funcionalidad se encuentra detallada en el Manual de Usuario.

Fig. 19: Formulario de postulación a oferta.

3.3.4 Configuración del método *get* para obtención de postulaciones de usuarios estudiantes y visualización de componente.

Dentro de la interfaz de perfil de usuario que mantiene la funcionalidad principal de la Aplicación Web se encuentra un componente el cual hace el llamado a la base de datos para obtener las postulaciones realizadas por el usuario que se encuentra autenticado como se muestra en la **Fig. 20**, dicho componente tiene los atributos necesarios para información del usuario, los atributos y su finalidad se encuentran detallados en el Manual de Usuario.



Mensaje de Postulación	Nombre de la empresa	Estado de postulación
Buenas noches, cuento con lo necesario para poder postular a su oferta.	Dicomsa	Por Contactar

Fig. 20: Componente de visualización de postulaciones.

3.3.5 Diseño del componente y su configuración para visualización de postulaciones de usuario estudiante y postulaciones a publicaciones de usuarios empresariales.

La **Fig. 21** ilustra el componente de visualización de postulaciones a las publicaciones que ha realizado el usuario empresarial que tiene sesión activa, los atributos que muestra el componente permiten al usuario empresarial realizar actividades como visualización de perfil de postulante y modificación del estado de la postulación del estudiante, de tal manera que permite tener al postulante una retroalimentación de la condición en la que se encuentra su solicitud a la oferta realizada por la empresa.

REALIZAR PUBLICACIÓN		VER POSTULACIONES			
Curriculum	Nombre de Postulante	Apellido de Postulante	Mensaje	Descripción de mi publicación	Estado de postulación
VER PERFIL	Gales	Guerrero	Tengo los conocimientos en los cuales se interesa.	Se solicita practicante con conocimientos en bases de datos SQL y NoSQL.	Por Contactar

Fig. 21: Componente de visualización de postulaciones.

La definición de los atributos del componente de postulaciones se encuentra detallado en el Manual de Usuario.

3.3.6 Diseño de la opción de visualización de perfiles de usuario estudiante.

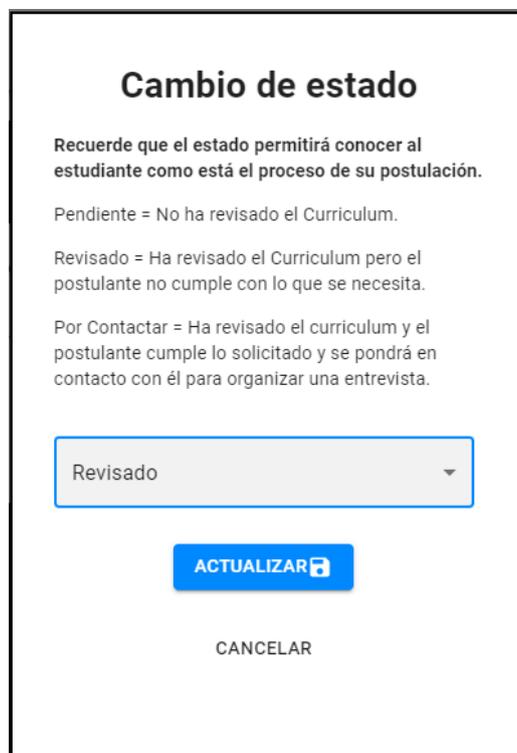
Los usuarios con cuenta empresarial cuentan con la opción de visualizar el perfil del usuario estudiante que postula a una de sus publicaciones, dicho perfil contiene el curriculum del postulante, de tal manera que el usuario con rol empresarial puede ver a detalle la lista de conocimientos que posee el estudiante y así concluir si dicha solicitud es adecuada para las necesidades que solicita, la **Fig. 22** muestra la interfaz perfil de estudiante tipo curriculum.



Fig. 22: Interfaz de perfil tipo curriculum.

3.3.7 Diseño del componente para actualización del estado de postulación.

Para mantener al usuario con rol estudiante al tanto sobre el estado de su postulación, el usuario con rol empresarial puede actualizar el estado de postulación de este, dicho componente cuenta con tres estados posibles como lo señala la **Fig. 23**, los cuales son: “Pendiente” que es el que se envía por defecto cuando se realiza una postulación, “Revisado” que da a indicar que se ha examinado la solicitud del estudiante pero que la empresa no se encuentra interesada en dicho perfil y por último, “Por Contactar” que da a entender que se ha revisado la solicitud del postulante y que la empresa está interesada en contratar dicho perfil.



Cambio de estado

Recuerde que el estado permitirá conocer al estudiante como está el proceso de su postulación.

Pendiente = No ha revisado el Curriculum.

Revisado = Ha revisado el Curriculum pero el postulante no cumple con lo que se necesita.

Por Contactar = Ha revisado el curriculum y el postulante cumple lo solicitado y se pondrá en contacto con él para organizar una entrevista.

Revisado

ACTUALIZAR

CANCELAR

Fig. 23: Componente de actualización de estado.

3.3.8 Diseño del componente para filtrado de publicaciones y ruta en *Back-end*.

Debido a la cantidad de ofertas que pueden ser publicadas por los usuarios con rol empresarial, el usuario estudiante puede filtrar las publicaciones por categoría como se indica en la **Fig. 24**, dicha categoría se encuentra almacenada en una entidad destinada para este objetivo, las categorías que se presenta se han definido previamente, debido al alcance que pueden tener dependiendo del tipo de practicante que solicita la empresa.

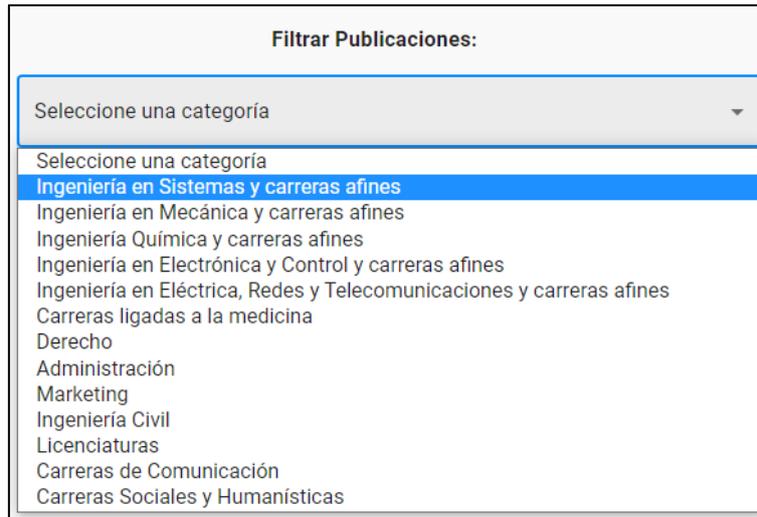


Fig. 24: Componente de filtrado de publicaciones.

Para la funcionalidad del componente se define una ruta que hace el llamado a la función en el controlador de la entidad *publications* para filtrar aquellas que contengan el identificador de la categoría que el usuario selecciona.

3.3.9 Configuración de visualización de resultados.

El componente en el que se encuentran las publicaciones está validado para que filtre aquellas que contengan la categoría que el usuario selecciona para visualizar, de esta manera como se muestra en la **Fig. 25** el componente se renderiza trayendo aquellas publicaciones de la base que cumplan con la condición de filtrado.

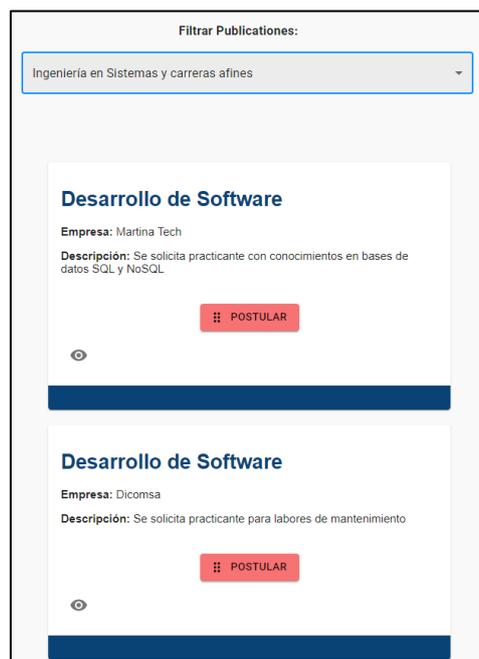


Fig. 25: Publicaciones - resultado de filtrado de publicaciones.

3.3.10 Implementación de la interfaz tipo foro para visualización y redacción de comentarios.

La **Fig. 26** indica la interfaz tipo foro, en la cual los usuarios que se encuentren con sesión activa pueden llenar el campo señalado para realizar un comentario, de igual manera pueden visualizar los comentarios realizados por otros usuarios junto con el nombre del creador de estos, sin embargo si el usuario no tiene sesión activa, el componente se encarga de realizar esta validación y oculta la opción de visualizar el formulario para el registro de un nuevo comentario.

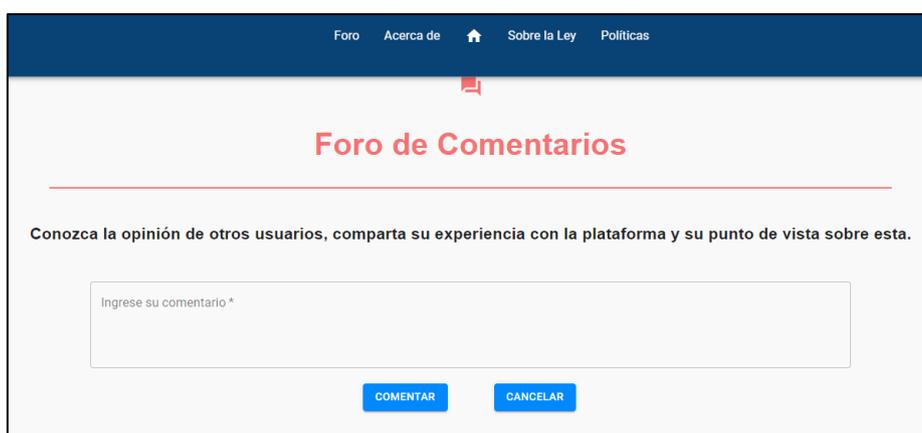


Fig. 26: Interfaz de foro.

3.3.11 Configuración de los recursos para obtención de usuario creador de comentario.

En el *Back-end* se configuran los recursos necesarios para la visualización del nombre y apellido del usuario que registra un comentario de tal manera que permita configurar el componente para que se visualice todo el detalle del comentario, el recurso queda como se muestra en la **Fig. 27**.

```
return [
  ... 'id' => $this->id,
  ... 'content' => $this->content,
  ... 'user' => '/api/users/' . $this->user_id,
  ... 'name' => $this->user->name,
  ... 'last_name' => $this->user->last_name,
  ... 'created_at' => $this->created_at,
  ... 'updated_at' => $this->updated_at,
];
```

Fig. 27: Configuración de recursos de entidad *Comments*.

3.4 *Sprint* 3. Implementación del módulo del usuario Administrador – Implementación del módulo de contacto con el administrador.

Siguiendo la planificación establecida en el *Sprint backlog* detallado en el Manual Técnico, a continuación, se especifican las actividades a realizar que corresponden al *Sprint* 3:

- Implementación de la interfaz de control del Administrador.
- Formulario y validación de los campos para registro de datos de un nuevo usuario en la base.
- Visualización de comentarios para eliminar y configuración del método *delete* para elidir dicho registro de la base.
- Diseño del contacto con el administrador - desarrollador dentro del componente *footer* y configuración del correo del desarrollador para recibir los mensajes junto con la validación de los campos a llenar.
- Visualización de alerta para envío de correo.

3.4.1 Implementación de la interfaz de control de Administrador.

La **Fig. 28** detalla los componentes que contiene la interfaz de gestión de actividades del usuario Administrador previamente registrado en la base por el desarrollador, en dicha interfaz se hace el llamado a los componentes para la gestión de ciertas actividades en la aplicación, las cuales en el Manual de Usuario se detallan.



Fig. 28: Interfaz administrativa.

3.4.2 Validación de los campos para registro de datos de nuevo usuario en la base.

El usuario Administrador tiene los permisos definidos en el *Back-end* para que pueda hacer el registro de un nuevo usuario, dicha funcionalidad contiene dos tipos de formularios que pueden ser llenados dependiendo del tipo de rol que dé el administrador al nuevo usuario, dicho componente se encuentra ilustrado en la **Fig. 29**.

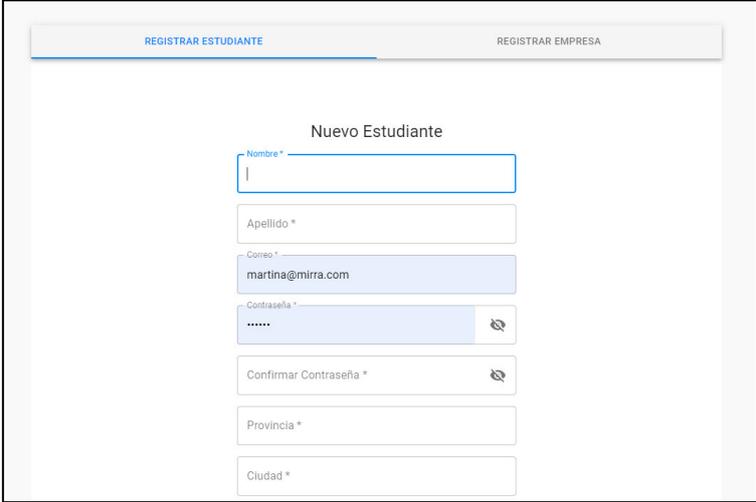


Fig. 29: Componente de registro de nuevos usuarios.

3.4.3 Visualización de comentarios para eliminar y configuración del método *delete* para elidir dicho registro de la base.

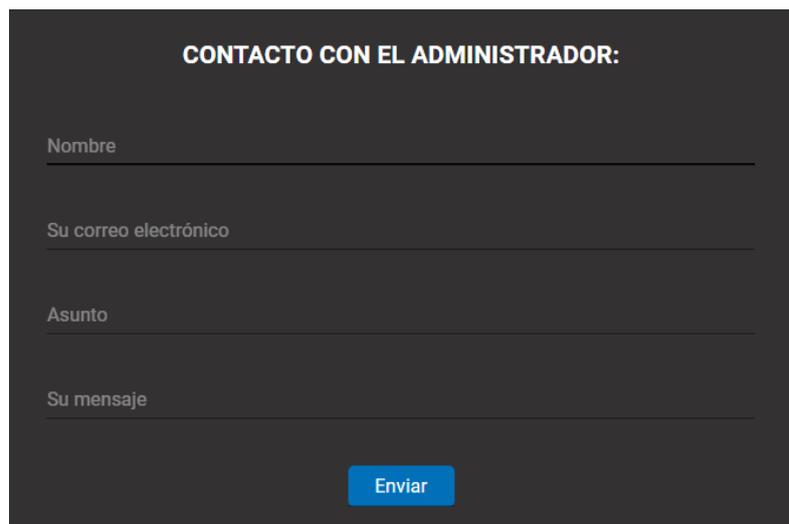
El componente que se muestra en la **Fig. 30**, permite al administrador visualizar el nombre del usuario que ha realizado dicho registro de comentario, de igual manera se muestra el cuerpo del comentario para validación u observación del administrador, también cuenta con un apartado de acción que hace el llamado al método *delete* del controlador de comentarios del *API*.

Nombre	Apellido	Texto	Acción
Martina	Mirra	En la parte de abajo de la página la parte de enviar un mensaje al desarrollador debería tener un indicador que muestre que se envió dicho mensaje.	Eliminar
Guillermo	Rivera	Conuerdo contigo Martina la combinación de colores es sobria	Eliminar

Fig. 30: Componente de eliminación de comentario.

3.4.4 Diseño del contacto con el administrador - desarrollador dentro del componente *footer* y configuración del correo del desarrollador.

La **Fig. 31** ilustra el formulario de contacto que se encuentra en el componente *footer* de la Aplicación Web, el cual permite llenar los campos simulando un medio para envío de correos, el objetivo del componente es recibir las quejas, consejos y consultas que los usuarios que visitan la Aplicación Web puedan tener para el administrador – desarrollador.



El formulario, con un fondo oscuro, tiene el título "CONTACTO CON EL ADMINISTRADOR:" en la parte superior. Incluye cuatro campos de entrada con el texto gris "Nombre", "Su correo electrónico", "Asunto" y "Su mensaje" respectivamente. En la parte inferior derecha del formulario hay un botón azul con el texto "Enviar".

Fig. 31: Componente de registro de nuevos usuarios.

Para el funcionamiento del componente se utiliza la herramienta *Email.js* el cual es un servicio para JavaScript para envío de correos por medio de un formulario y hace uso de credenciales provenientes de los servicios de mensajería [24].

3.4.5 Visualización de alerta para envío de correo.

La **Fig. 32** detalla la alerta que se muestra en pantalla al momento de que se ha enviado el correo del usuario, esta funcionalidad está incorporada de manera pública lo que involucra el libre acceso al uso de ella.

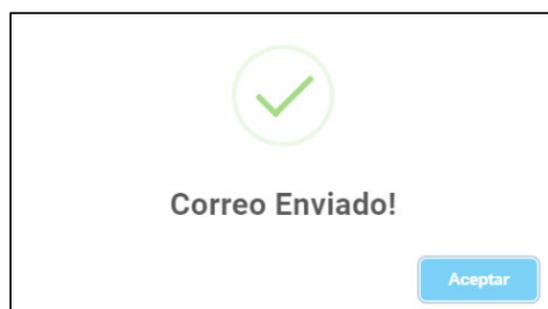


Fig. 32: Alerta de comprobación de envío de correo.

3.5 *Sprint* 4. Implementación de módulos de visualización de perfiles de usuarios y descripción de Ley – Implementación de alertas y detalle de publicación.

A continuación, prosiguiendo con las actividades establecidas en el *Sprint backlog* que se encuentra en el Manual Técnico, se especifican las tareas a realizar que corresponden al *Sprint* 4:

- Implementación de la Interfaz de descripción de la Ley ecuatoriana.
- Configuración de plugin para mejor experiencia de usuario con las alertas.
- Configuración del método *post* para visualización detallada de la publicación por parte del usuario con rol empresarial.

3.5.1 Implementación de la interfaz de descripción de la Ley.

En vista de que el objetivo de la aplicación son las ofertas de prácticas pre profesionales, es importante que los usuarios conozcan lo que estipula el reglamento y la Ley ecuatoriana con respecto a este tipo de actividades, de esta manera los usuarios finales entienden el porqué de la Aplicación Web y sus funcionalidades, dicha interfaz cuenta con un componente y un enlace que redirige al documento como se ilustra en la **Fig. 33**, donde se detalla lo que involucra las horas de prácticas pre profesionales para los usuarios que quieran conocer más acerca del tema.



Fig. 33: Interfaz referente a la Ley Ecuatoriana.

3.5.2 Configuración del método *get* para visualización de los perfiles de usuarios estudiantes.

Debido a la solicitud de visualización de perfil, es necesario la configuración de una ruta en el *Back-end* que haga el llamado al controlador para que permita la visualización del usuario que postula a una publicación del usuario de rol empresarial que se encuentra con sesión activa, la **Fig. 34** señala la función en el controlador para la acción de visualización.

```
public function showUserCurriculum(User $user)
{
    $this->authorize('viewUserCurriculum', User::class);
    $curriculum = CVitae::where('user_id', $user['id'])->get();
    return response()->json(new CVitaeCollection($curriculum), 200);
}
```

Fig. 34: Función para visualización del curriculum.

3.5.3 Configuración de *plugin* para mejor experiencia de usuario con las alertas.

La **Fig. 35** señala la página del *plugin SweetAlert*, el cual brinda la posibilidad de dar un aspecto diferente a los mensajes de alerta que pueda tener la Aplicación Web [25], de tal manera que posibilita mejorar la experiencia del usuario durante el consumo de la Aplicación Web y las acciones que permite esta.

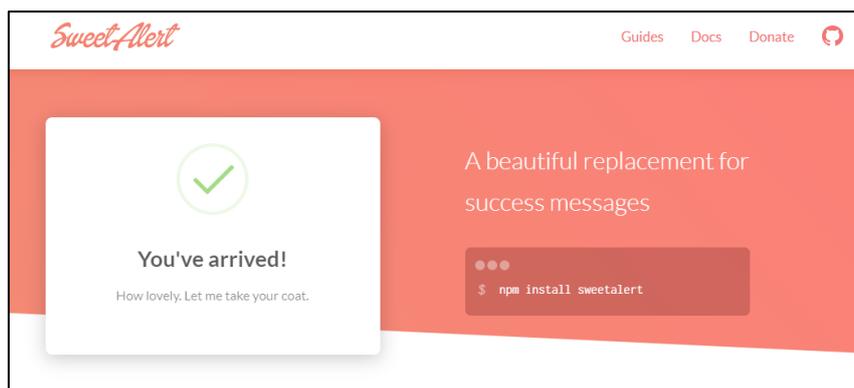
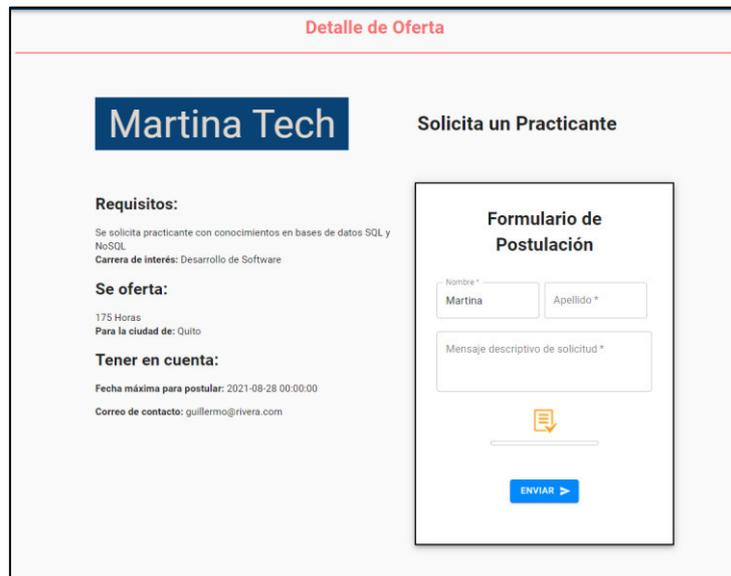


Fig. 35: Página de plugin *SweetAlert*.

3.5.4 Configuración del método *post* para visualización detallada de la publicación realizada por parte del usuario con rol empresarial.

Dentro de las actividades que puede realizar el usuario con rol estudiantil, está permitida la verificación a detalle de una oferta de prácticas pre profesionales, la funcionalidad se desarrolla dentro de un componente que hace el llamado a la ruta de publicaciones y

específicamente al identificador de la publicación de la cual se desea visualizar a detalle, por tal caso, para la obtención del resultado se ha implementado el uso de *props* para el llamado y obtención de datos entre los componentes, la **Fig. 36** muestra la interfaz de detalle de una publicación seleccionada.



The screenshot shows a web interface titled "Detalle de Oferta" (Job Offer Detail). On the left, there is a logo for "Martina Tech" and a section titled "Solicita un Practicante" (Apply for an Intern). Below this, there are three sections: "Requisitos:" (Requirements) stating the need for SQL/NoSQL knowledge and software development; "Se oferta:" (Offered) listing 175 hours and the city of Quito; and "Tener en cuenta:" (Keep in mind) with a deadline of 2021-08-28 00:00:00 and contact email guillermo@rivera.com. On the right, there is a "Formulario de Postulación" (Application Form) with fields for "Nombre *" (Name) containing "Martina" and "Apellido *" (Last Name), a "Mensaje descriptivo de solicitud *" (Descriptive message of application) field, and a blue "ENVIAR" (SEND) button.

Fig. 36: Visualización de detalle de oferta.

3.6 *Sprint 5*. Implementación de pruebas a la Aplicación Web – Despliegue completo de la Aplicación Web.

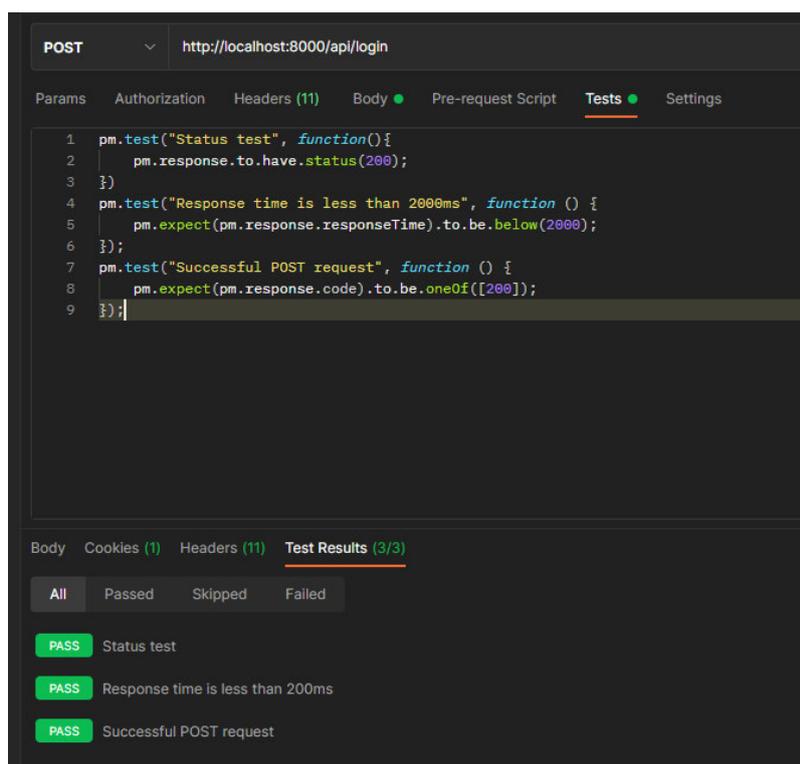
Posterior al desarrollo de las actividades señaladas en los *Sprints* anteriores adjuntos en el Manual Técnico, a continuación, se establecen las actividades que involucran las pruebas a la Aplicación Web, permitiendo de esta manera el aseguramiento del correcto funcionamiento de los módulos y componentes que representan la estructura de la Aplicación Web, de tal manera que se alcance el objetivo de calidad, además, realizadas las pruebas se mencionan las acciones de subida al servidor de las partes *Back-end* y *Front-end*

- Pruebas unitarias
- Pruebas de usuario.
- Pruebas de rendimiento.
- Despliegue de la *API* en el servidor *Heroku*.
- Despliegue de la aplicación en el host de *Vercel*.

3.6.1 Pruebas Unitarias.

Las pruebas unitarias son un tipo de método que comprueba una parte de código, al realizarse por medio de pequeñas cantidades de código dan resultados de manera rápida y detallan el estado de funcionamiento de la colección de líneas a las cuales sean aplicado la prueba, este tipo de pruebas son independientes a otros componentes por lo cual el resultado que devuelven corresponde únicamente a la parte en la que se incorpora la prueba, cabe mencionar que estas pruebas no inciden en el rendimiento de la Aplicación Web por lo cual la cantidad de veces que se la implemente a una misma porción de código no significa un problema [26].

A continuación, en la **Fig. 37** se muestra el resultado de la prueba unitaria implementada al método *post* de la ruta de inicio de sesión de la Aplicación Web, para la implementación de la prueba se hace uso de la herramienta *test* que ofrece *postman*, en la cual es necesario el ingreso de las líneas de código, de tal manera que devuelva el estado de la petición.



```
1 pm.test("Status test", function(){
2   pm.response.to.have.status(200);
3 })
4 pm.test("Response time is less than 2000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(2000);
6 });
7 pm.test("Successful POST request", function () {
8   pm.expect(pm.response.code).to.be.oneOf([200]);
9 });
```

Body Cookies (1) Headers (11) **Test Results (3/3)**

All	Passed	Skipped	Failed
PASS	Status test		
PASS	Response time is less than 200ms		
PASS	Successful POST request		

Fig. 37: Prueba 1 - Módulo de inicio de sesión.

Las pruebas unitarias se detallan en la sección del Manual Técnico – Pruebas unitarias página 35, que concluyen que las rutas involucradas en la parte del *API* generan resultados exitosos.

3.6.2 Pruebas de usuario.

Para la implementación de las pruebas de usuario, se ha seleccionado un grupo reducido de posibles usuarios finales de la Aplicación Web para que hagan uso y consumo de esta. Para la elaboración de dicha prueba se especifican una serie de preguntas realizadas a los usuarios al final de la interacción con la Aplicación Web, de tal forma, que los resultados dan una idea retroalimentativa con respecto a su experiencia a la hora de uso, el resultado de estas pruebas es detallado en la sección del Manual Técnico – Pruebas de usuario página 39.

3.6.3 Pruebas de rendimiento.

Estas pruebas son un proceso que involucra una alta gestión y compromiso debido a que precisa el estado de respuesta que tiene la Aplicación Web concerniente a las peticiones que realizan los usuarios, la prueba se desarrolla con ayuda de la herramienta JMeter, la cual permite realizar *testing* de rendimiento a servicios web [27], aportando así con un *feedback* de estado de la Aplicación Web con concurrencia de usuarios y datos.

La **Fig. 38** ilustra el uso de la herramienta JMeter para la prueba de rendimiento implementada a la petición de obtención de comentarios.

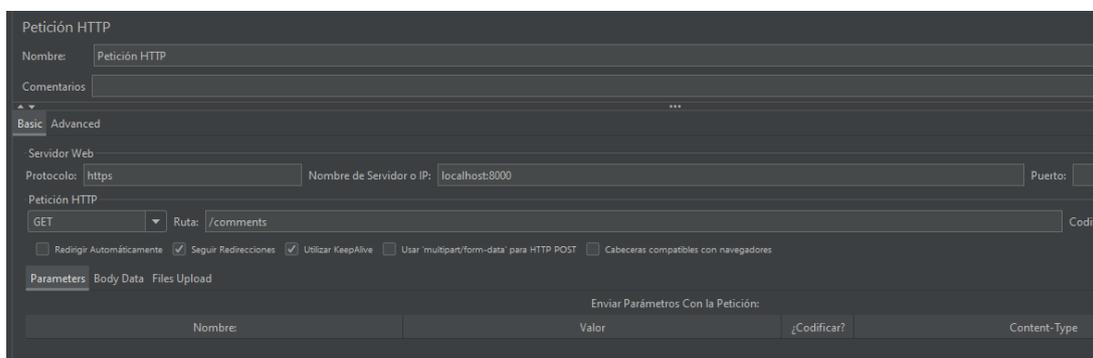


Fig. 38: Implementación de JMeter.

Los resultados devueltos por las pruebas son exitosos, variando por aquellas que involucran un alto grado de estrés que han denotado una funcionalidad estable en este tipo de situaciones, las pruebas se detallan en la sección Manual Técnico – Pruebas de rendimiento página 44.

3.6.4 Despliegue de la API en servidor Heroku.

Terminado el proceso de desarrollo, adecuación de la base de datos, componentes, módulos y pruebas al *Back-end* y *Front-end* de la Aplicación Web, se realiza el debido

despliegue del *API* en un servidor para poder consumirla desde la Aplicación Web, este despliegue se implementa por medio de *Heroku*, que permite guardar en la nube una aplicación tipo *Rest* dando como resultado el almacenaje de los registros en una base de datos en *PostgreSQL* que se implementa automáticamente durante el despliegue, por lo cual, al finalizar se obtiene una *url* del proyecto con un protocolo seguro como lo es *https*.

La **Fig. 39** muestra la primera etapa de subida del *API* en la nube, por otro lado, la **Fig. 40** indica que el despliegue de la aplicación se ha realizado con éxito y ya se puede hacer consultas y consumo a la *url* que retorna el despliegue del proyecto.

```
G:\UNIVERSIDAD\TESIS\PARTE PRÁCTICA\BackEnd_TESIS>heroku create
» Warning: heroku update available from 7.53.0 to 7.59.0.
Creating app... done, ⬢ thawing-bastion-41234
https://thawing-bastion-41234.herokuapp.com/ | https://git.heroku.com/thawing-bastion-41234.git

G:\UNIVERSIDAD\TESIS\PARTE PRÁCTICA\BackEnd_TESIS>heroku push
» Warning: heroku update available from 7.53.0 to 7.59.0.
» Warning: push is not a heroku command.
Did you mean ps? [y/n]: y
Free dyno hours quota remaining this month: 550h 0m (100%)
Free dyno usage for this app: 0h 0m (0%)
For more information on dyno sleeping and how to upgrade, see:
https://devcenter.heroku.com/articles/dyno-sleeping

No dynos on ⬢ thawing-bastion-41234

Setting APP_NAME and restarting ⬢ thawing-bastion-41234... done, v3
APP_NAME: Laravel

G:\UNIVERSIDAD\TESIS\PARTE PRÁCTICA\BackEnd_TESIS>heroku config:add APP_ENV=production
» Warning: heroku update available from 7.53.0 to 7.59.0.
Setting APP_ENV and restarting ⬢ thawing-bastion-41234... done, v4
APP_ENV: production
```

Fig. 39: Creación y configuración del *deploy* del *API*.

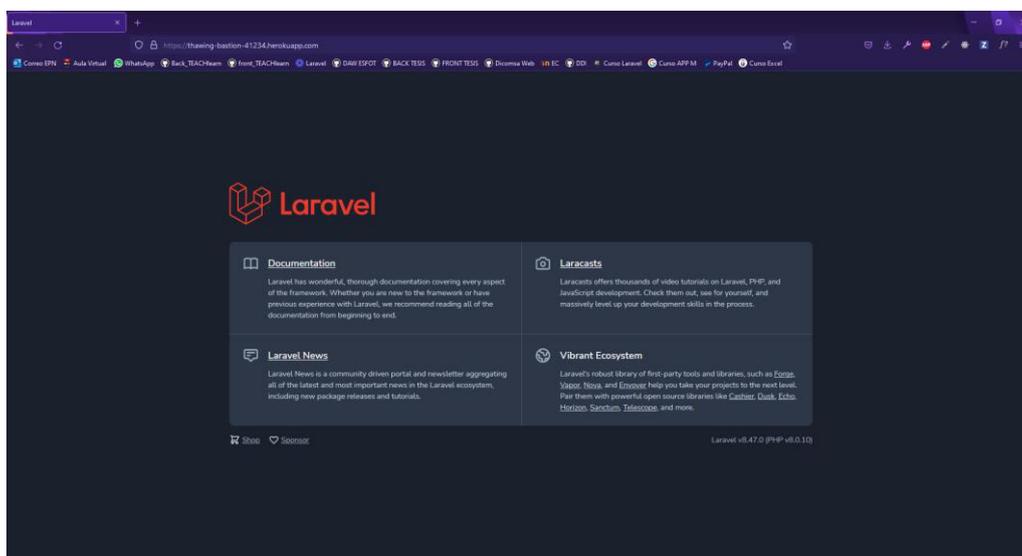


Fig. 40: Despliegue del *API* en *Heroku*.

Todo el proceso de subida a la nube se encuentra detallado en la sección del Manual Técnico – Despliegue del *API* en *Heroku* página 46.

3.6.5 Despliegue de la aplicación en el *host* de Vercel.

Si bien el proceso de subida a la nube del *Back-end* de la Aplicación Web permite realizar consultas de manera externa al localhost, es pertinente continuar con el proceso de *deploy* del *Front-end* de la Aplicación Web, en este caso se hace uso del servidor *Vercel* que proporciona tanto *host* como dominios gratuitos para la subida de este tipo de servicios *web*.

La **Fig. 41** muestra la primera etapa de subida de la aplicación al servidor de *Vercel*.

La Aplicación Web se encuentra en la siguiente *url*: **https://practicas-al-dia.vercel.app/**

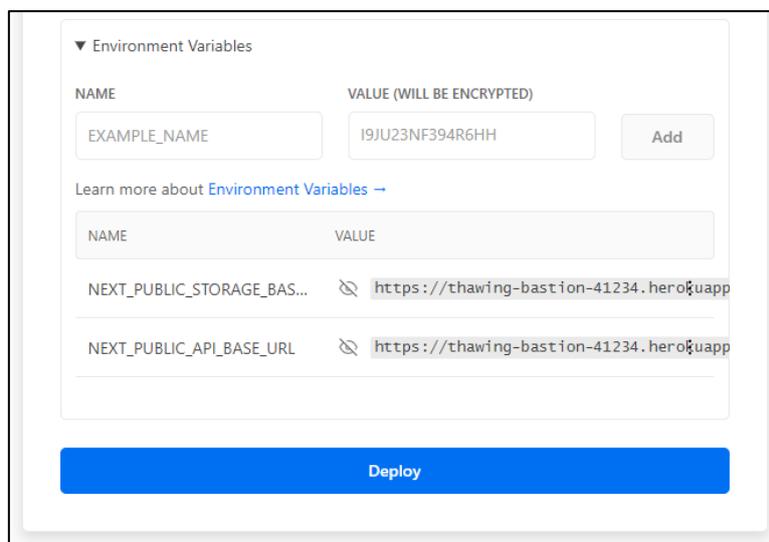


Fig. 41: Configuración de variables de entorno.

Por otro lado, la **Fig. 42** muestra el resultado del despliegue de la Aplicación Web en el servidor, retornando tres *url* que permiten su visualización.

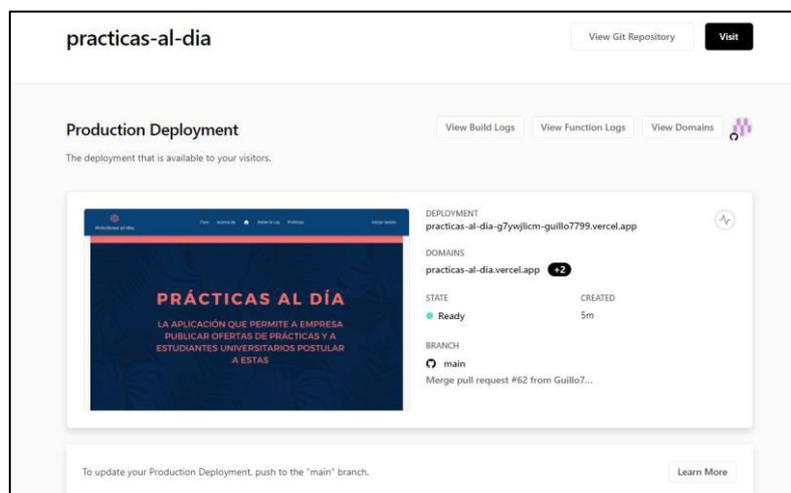


Fig. 42: Resultado del *deploy* de la Aplicación Web en *Vercel*.

Todo el proceso de *deploy* a los servidores se encuentra detallado en la sección del Manual Técnico – Despliegue de la Aplicación Web en Vercel página 55.

4 CONCLUSIONES Y RECOMENDACIONES

A continuación, se listan un conjunto de conclusiones y recomendaciones que se han ido obteniendo en el proceso de desarrollo de la Aplicación Web.

4.1 Conclusiones

- La Aplicación Web cumple con el alcance, objetivos y necesidades previamente establecidos en el levantamiento de requerimientos, de tal manera que cada componente y/o módulo cuenta con las características que permiten realizar la determinada acción por las cuales fueron planteadas y desarrolladas.
- Los estudiantes universitarios que necesitan cumplir con este requisito previo a la obtención de su título pueden realizar postulaciones a ofertas de prácticas dentro la Aplicación Web que cuenta con una interfaz interactiva y fácil de aprender, considerando la necesidad de un registro previo a la gestión de las actividades a realizar dentro de la Aplicación Web.
- El marco de trabajo ágil *Scrum* permite enfocarse en objetivos propuestos de manera iterativa, produciendo riesgos mínimos y nulos enfocados a retrasos en el proceso de desarrollo de la Aplicación Web, por lo tanto, su implementación posibilita un enfoque de cumplimiento de metas específicas por *Sprint*, que generan un incremento en el producto final.
- La implementación de librerías y *plugins* de *UI* para mejorar la experiencia de uso, dan como resultado una Aplicación Web fácil de entender con respecto al objetivo que propone, amigable a la vista del usuario con una interacción sencilla que son las principales características de un proyecto que está dirigido a la calidad de software, por consiguiente, genera un incentivo y llama la atención de posibles usuarios finales a que hagan uso de las funciones que contiene.
- Los servicios de *hosting* utilizados para ambas partes que constituyen la Aplicación Web dan un agregado de seguridad y alcance de esta a cualquier tipo de usuario, esto debido al manejo de protocolos de seguridad que objetan que los datos ingresados son manejados de forma segura y no expuestos a terceros, este tipo de detalles se resaltan en las políticas de uso y privacidad que pertenecen a un apartado específico de la Aplicación Web, sin embargo, hay que tomar en cuenta que el servidor de *Heroku* implementa la base de datos en *PostgreSQL* de forma automática, que no se detalla dentro de las herramientas utilizadas, aunque, contiene las mismas características y funcionalidades que la base de datos en *MySQL*, que se menciona en el presente informe.

- Las pruebas realizadas a la Aplicación Web presentaron resultados que permiten comprobar su correcto funcionamiento en los diferentes módulos y componentes testeados, cumpliendo los objetivos del ciclo de desarrollo del proyecto detallado en el presente informe.

4.2 Recomendaciones

- Como la Aplicación Web solicita el registro de datos sensibles de usuarios como lo son ubicaciones, correos y números de contacto, se recomienda realizar un análisis de vulnerabilidad cada cierto tiempo que la Aplicación Web haya sido utilizada, de tal manera que se siga asegurando los datos que transitan dentro de esta.
- Para una nueva versión de la Aplicación Web se recomienda agregar ciertas funcionalidades de edición de datos que no involucren violaciones en las restricciones de la base, de tal manera que el flujo continuo de consumo de datos y modificación de los mismo sea el correcto y no involucre fallos.
- Debido a que una de las acciones principales que realiza uno de los usuarios es la subida de datos personales y profesionales, se recomienda que estos al momento de ser registrados sean actuales, así pues, que el proceso de postulación de dichos usuarios sea el correcto y oportuno dependiendo del caso de oferta.
- Tomando como punto de referencia las pruebas de usuarios realizadas, se recomienda que los usuarios finales visualicen las interfaces que corresponden a la explicación de las políticas de uso y privacidad referentes a la Aplicación Web y también a un apartado conciso acerca de lo que involucra la ley ecuatoriana en las prácticas pre profesionales.
- Debido al uso de una herramienta gratuita como lo es *Heroku* para el almacenamiento del *API* no se puede manejar un correcto renderizado de la actualización de datos del usuario en tiempo real, por lo cual, se recomienda en futuras versiones de la Aplicación Web implementar dicha característica con herramientas privadas que otorguen un mejor servicio.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] «Universia,» 11 06 2020. [En línea]. Available: <https://www.universia.net/pe/actualidad/empleo/que-son-practicas-pre-profesionales-aqui-te-lo-contamos-845583.html>. [Último acceso: 05 02 2021].
- [2] Careerjet, «Opcionempleo,» 2021. [En línea]. Available: <https://www.opcionempleo.ec/>. [Último acceso: 04 06 2021].
- [3] R. P. y. E. Sobakarev, «Joooble,» 2006. [En línea]. Available: <https://ec.jooble.org/>. [Último acceso: 04 06 2021].
- [4] S. G. Crespo, «Dialnet,» 2018. [En línea]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=6778218>. [Último acceso: 31 01 2021].
- [5] R. M. d. Vega, «Profile,» 24 08 2020. [En línea]. Available: <https://profile.es/blog/desarrollo-aplicaciones-web/>. [Último acceso: 04 06 2021].
- [6] G. SBOK, «Ti Think,» 16 10 2018. [En línea]. Available: <https://www.tithink.com/es/2018/10/16/metodologias-agiles-que-son-y-para-que-sirven/>. [Último acceso: 04 06 2021].
- [7] A. Djandrw, «Andrew Djandrw,» Medium, 13 04 2019. [En línea]. Available: <https://medium.com/@andrewdjandrw/qu%C3%A9-es-Scrum-674c6b791af4>. [Último acceso: 02 05 2021].
- [8] «Softeng,» Softeng your competitive advantage, [En línea]. Available: <https://www.softeng.es/es-es/empresa/metodologias-de-trabajo/metodologia-Scrum/proceso-roles-de-Scrum.html>. [Último acceso: 04 06 2021].
- [9] A. Pérez, «OBS Business School,» 15 07 2016. [En línea]. Available: <https://www.obsbusiness.school/blog/principales-roles-de-la-metodologia-agil-Scrum>. [Último acceso: 04 06 2021].
- [10] A. G. Ochoa, «Agileando,» [En línea]. Available: <https://agileando.com/que-son-los-artefactos-de-Scrum/>. [Último acceso: 04 06 2021].

- [11] «*Scrum* México,» 02 08 2018. [En línea]. Available: <https://Scrum.mx/informate/historias-de-usuario>. [Último acceso: 04 06 2021].
- [12] «Proyectos Ágiles,» [En línea]. Available: <https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog/>. [Último acceso: 04 06 2021].
- [13] M. García, «It Tude,» 17 07 2020. [En línea]. Available: <https://ittude.com.ar/b/Scrum/que-es-el-Sprint-backlog/>. [Último acceso: 04 06 2021].
- [14] C. Bravo, «EstudioKa,» [En línea]. Available: <https://estudioka.es/que-es-un-mock-up/>. [Último acceso: 20 06 2021].
- [15] «ISDI Talento digital,» 02 10 2014. [En línea]. Available: <https://www.isdi.education/es/blog/balsamiq-herramienta-para-realizar-prototipos-de-tus-proyectos>. [Último acceso: 20 06 2021].
- [16] J. Casanovas, «Desarrolloweb.com,» 09 09 2004. [En línea]. Available: <https://desarrolloweb.com/articulos/1622.php>. [Último acceso: 22 06 2021].
- [17] «IBM,» IBM Corp, 2006. [En línea]. Available: https://cgrw01.cgr.go.cr/rup/RUP.es/SmallProjects/core.base_rup/guidances/termindefiniciones/architectural_pattern_E2E8EB79.html. [Último acceso: 22 06 2021].
- [18] «Nodejs,» OpenJs Foundation, [En línea]. Available: <https://nodejs.org/es/about/>. [Último acceso: 25 06 2021].
- [19] A. S. E. Y. CRM, «El blog de Aitana,» 16 10 2018. [En línea]. Available: <https://blog.aitana.es/2018/10/16/visual-studio-code/>. [Último acceso: 25 06 2021].
- [20] M. García, «Nettix,» 30 05 2020. [En línea]. Available: <https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo>. [Último acceso: 25 06 2021].
- [21] A. López, «Open Webinars,» 03 06 2019. [En línea]. Available: <https://openwebinars.net/blog/que-es-postman/>. [Último acceso: 25 06 2021].
- [22] AminuddinB., «Capterra,» 29 01 2021. [En línea]. Available: <https://www.capterra.es/software/203626/vercel>. [Último acceso: 25 06 2021].

- [23] J. L. Coalla, «Trybalite Technologies,» 23 06 2021. [En línea]. Available: <https://tech.tribalyte.eu/blog-que-es-react>. [Último acceso: 19 09 2021].
- [24] H. Guerrero, «HOC Labs,» 17 03 2018. [En línea]. Available: <https://www.hoclabs.com/2018/03/17/emailjs-envio-de-correos-desde-javascript/>. [Último acceso: 30 08 2021].
- [25] T. E. y. Contribuidores, «SweetAlert,» [En línea]. Available: <https://sweetalert.js.org/>. [Último acceso: 30 08 2021].
- [26] A. Sommariva, «MicroGestion,» [En línea]. Available: <http://www.microgestion.com/index.php/mg-developers/articulos/74-unit-test-part1-mock>. [Último acceso: 30 08 2021].
- [27] «IBM,» [En línea]. Available: https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/guidances/concepts/acceptance_testing_12A0F152.html. [Último acceso: 30 08 2021].
- [28] J. Cabana, «Drauta,» 01 08 2017. [En línea]. Available: <https://www.drauta.com/que-es-nodejs-y-para-que-sirve>. [Último acceso: 25 06 2021].

6 ANEXOS

Seguidamente, se presentan los anexos concernientes al presente informe del desarrollo de la Aplicación Web.

6.1 Manual Técnico

- Recopilación de requerimientos.
- Historias de Usuario.
- *Product Backlog*.
- *Sprint Backlog*.
- Diseño de Interfaces.
- Diseño de la Base de datos.
- Pruebas de usuario.
- Pruebas unitarias.
- Pruebas de rendimiento.
- Pruebas de Integración.
- Despliegue del *API* en *Heroku*.
- Despliegue de la Aplicación Web en *Vercel*.

6.2 Manual de Usuario

A continuación, se presenta un *link* general que contiene a los repositorios de la Aplicación Web, tanto *Back-end* y *Front-end*, además del enlace del video en YouTube acerca del funcionamiento.

Repositorio: https://github.com/Guillo7799/Tesis_RiveraGuillermo