

UNIVERSIDAD POLITÉCNICA DE PUEBLA
Ingeniería en Tecnologías de la Información



Proyecto de Estadía Profesional

“Sistema web para la gestión del
Laboratorio Lerim por parte de
FrontEnd”

Área temática del CONACYT: VII
Ingenierías y tecnologías

Presenta:

Samuel Martínez Arenas

Asesor técnico

Ingeniero David Ángel Hernández Sánchez

Asesor académico

Dra. Araceli Ortiz Carranco

Juan C. Bonilla, Puebla, México.

Fecha fin de cuatrimestre

Resumen

El presente informe detalla el proceso, la documentación y los resultados obtenidos durante la realización del proyecto de Estadía correspondiente al décimo cuatrimestre de la Ingeniería en Tecnologías. Este proyecto se llevó a cabo en la empresa CodeLines, ubicada en la colonia Valle Dorado, Puebla.

La iniciativa se centró en el desarrollo de un sistema web por parte del frontend, destinado a la gestión del laboratorio microbiológico Lerim. Con el objetivo de alcanzar este propósito, se optó por la utilización de tecnologías de programación, como Laravel y React. A lo largo de este informe, se presentan de manera detallada las diferentes etapas del proyecto, siguiendo la metodología ágil de desarrollo de software conocida como Scrum.

Índice

1	Introducción.....	8
1.1.	Descripción del problema o necesidad.....	8
1.2.	Justificación.....	8
1.3.	Objetivo General y Específicos	9
1.3.1.	Objetivo General:.....	9
1.3.2.	Objetivos Específicos:	9
2	Metodología y herramientas.....	9
2.1.	Metodología Scrum	9
2.1.1.	Fases procesos de Scrum	10
2.1.2.	Ventajas y desventajas	12
2.2.	Herramientas.....	13
2.2.1.	Visual Studio Code	13
2.2.2.	Postman	14
2.2.3.	PhpMyAdmin	15
2.2.4.	Xampp	15
2.2.5.	Composer	16
2.2.6.	React	17
2.2.7.	Node.js	18
2.2.8.	GitHub	19
2.2.9.	Laravel.....	20
2.2.10.	Google Chrome	20
3	Resultados	22
3.1	Iniciación	22
3.1.1	Visión del Proyecto	22
3.1.2	Formación del equipo Scrum	22
3.2	Planificación y estimación	22
3.3	Implementación.....	23
3.3.1	Sprint 1	23
3.4	Revisión y Retrospectiva.....	49
3.4.1	Revisión de Sprints.....	49

3.4.2	Retrospectiva de Sprints.....	50
3.5	Lanzamiento.....	50
3.5.1	Sprint 1	50
3.5.2	Sprint 2	51
3.5.3	Sprint 3	51
3.5.4	Sprint 4	51
3.5.5	Sprint 5	52
4	Conclusiones y recomendaciones.....	53
5	Referencias bibliográficas	54

Índice de figuras

Figura 1. Fases de Scrum	10
Figura 2. Instalación de paquetes en la plantilla.....	24
Figura 3. Interfaz de inicio de sesión	24
Figura 4. Código para inicio de sesión	25
Figura 5. Uso de la API.....	25
Figura 6. Código para crear un nuevo Usuario.....	26
Figura 7. Uso de la API para crear un nuevo Usuario.	26
Figura 8. Interfaz para crear un Nuevo Usuario.	27
Figura 9. Código para actualizar un Usuario.	27
Figura 10. Uso de la API para actualizar un Usuario.....	28
Figura 11. Interfaz para actualizar un Usuario.	28
Figura 12. Código para ver a los usuarios registrados.	29
Figura 13. Interfaz para ver a los usuarios registrados.....	29
Figura 14. Código para crear un nuevo Rol.....	30
Figura 15. Uso de la API para crear un nuevo Rol.	30
Figura 16. Interfaz para crear un nuevo Rol.	31
Figura 17. Código para actualizar un Rol.	31
Figura 18. Uso de la API para actualizar un Rol.....	32
Figura 19. Interfaz para actualizar un Rol.....	32
Figura 20. Código para ver a los roles registrados.	33
Figura 21. Interfaz para ver a los roles registrados.	33
Figura 22. Código para crear un nuevo Análisis.....	34
Figura 23. Uso de la API para crear un nuevo Análisis.....	34
Figura 24. Interfaz para crear un nuevo Rol.	35
Figura 26. Uso de la API para actualizar un Análisis.	36
Figura 27. Interfaz para actualizar un Análisis.	36
Figura 28. Código para ver a los análisis registrados.....	37
Figura 29. Interfaz para ver a los análisis registrados.	37
Figura 30. Código para crear un nuevo Cliente.	38
Figura 31. Uso de la API para crear un nuevo Análisis.....	38
Figura 32. Interfaz para crear un nuevo Cliente.	39

Figura 34. Uso de la API para actualizar un Cliente.	40		
Figura 35. Interfaz para actualizar un Cliente.	40		
Figura 36. Código para ver a los clientes registrados.....	41		
Figura 37. Interfaz para ver a los clientes registrados.	41		
Figura 38. Código para actualizar las Muestras.	42		
Figura 41. Código para ver las Muestras.....	44		
Figura 42. Interfaz para ver las Muestras.	44		
Figura 43. Código para crear las Ordenes de Trabajo. ¡Error!	Marcador	no	definido.
Figura 44. Interfaz para crear las Ordenes de Trabajo. ¡Error!	Marcador	no	definido.
Figura 45. Código para ver las Ordenes de Trabajo. ¡Error!	Marcador	no	definido.
Figura 46. Interfaz para ver las Ordenes de Trabajo. ¡Error!	Marcador	no	definido.
Figura 47. Interfaz para mandar los resultados de las muestras.	¡Error!	Marcador	no definido.
Figura 48. Interfaz para confirmar el envío de correo. ¡Error!	Marcador	no	definido.
Figura 50. Correo recibido.	¡Error!	Marcador	no definido.

Índice de Tablas

Tabla 1. Formación del equipo scrum.....	22
Tabla 2. Backlog	23
Tabla 3. Sprint 1.....	23
Tabla 4. Sprint 2.....	25
Tabla 5. Sprint 3.....	38
Tabla 6. Sprint 4.....	42
Tabla 7. Sprint 5.....	45
Tabla 8. Revisión de Sprints	50
Tabla 9. Retrospectiva de Sprints.....	50
Tabla 10. Verificación de Sprint 1	50
Tabla 11. Verificación de Sprint 2.....	51
Tabla 12. Verificación de Sprint 3	51
Tabla 13. Verificación de Sprint 4	51
Tabla 14. Verificación de Sprint 5	52

1 Introducción

En este capítulo se hablará sobre la descripción del problema, así como la justificación del desarrollo del proyecto que se realizó en la durante la estadía que se realizó en la empresa CodeLines.

1.1. Descripción del problema o necesidad

El laboratorio microbiológico Lerim necesita una aplicación web para gestionar y mostrar los resultados de las pruebas microbiológicas que realiza ya que no lleva un control de sus registros de las pruebas y análisis de los diferentes negocios a los que le trabajan.

La ausencia de un sistema web en el laboratorio da lugar a una serie de problemas y desafíos como lo son la ineficiencia de los datos, dificultad en la gestión y programación de las pruebas o análisis; y esto puede ocasionar que haya inconsistencias en los registros, que haya mala comunicación con los clientes o que no se puedan comprobar las normas y regulaciones que se les puedan solicitar.

Es por ello que la falta de un sistema en el laboratorio Lerim puede tener un impacto significativo en la eficiencia operativa, la calidad de servicio, la seguridad de datos y la capacidad de adaptarse a las necesidades cambiantes.

Un ejemplo de la ineficiencia de los datos que se pueden encontrar es la repetición de datos cuando se toman las diferentes muestras o los análisis y esto mismo puede causar confusión al hacer el llenado de los campos que se soliciten.

También la dificultad de la gestión y programación de las pruebas que se realizan es la confusión de horas y fechas de la recepción y entrega de las pruebas o al mismo tiempo la acumulación de solicitudes que puedan llegar al laboratorio y debido a la acumulación puede haber una mezcla de datos de un cliente con los de otro u otros clientes.

1.2. Justificación

La determinación de crear un sistema de gestión web orientado al frontend para el laboratorio Lerim se fundamenta en la imperante necesidad de superar los retos vinculados a la administración documental y la interacción con los clientes. La complejidad asociada a la manipulación manual de documentos físicos, especialmente en lo que respecta a la interpretación y entrega de resultados de análisis microbiológicos, impacta negativamente la eficiencia operativa y la calidad del servicio.

En este contexto, la propuesta se centra en la automatización de la generación de documentos, la mejora de la estructura de resultados y la facilitación de la entrega digital, con el objetivo de disminuir la carga laboral del personal y elevar la precisión en la comunicación con los clientes. La selección de Laravel y React

se basa en su efectividad y versatilidad, asegurando una arquitectura robusta y sostenible. Esta iniciativa no solo repercute positivamente en Lerim, sino que también establece un paradigma de eficiencia para laboratorios afines, contribuyendo al progreso y modernización en la gestión de servicios de análisis microbiológicos.

1.3. Objetivo General y Específicos

1.3.1. Objetivo General:

Desarrollar e implementar una aplicación web eficiente y amigable para usuarios, que permita a un laboratorio microbiológico gestionar los datos de las pruebas y análisis de manera efectiva, brindando una interfaz de usuario atractiva y funcional.

1.3.2. Objetivos Específicos:

1. Crear un diseño de interfaz de usuario intuitivo y atractivo que refleje la naturaleza científica y profesional del laboratorio microbiológico.
2. Diseñar una estructura de navegación fácil de entender y usar.
3. Asegurar una comunicación efectiva entre el front end y el backend de la aplicación, permitiendo la gestión de datos.
4. Implementar filtros y herramientas de búsqueda para facilitar la recuperación de información.
5. Incorporar reportes claros y legibles para presentar datos microbiológicos de manera efectiva.
6. Implementar un esquema de colores y tipografía coherente y agradable a la vista.

2 Metodología y herramientas

En este capítulo se presentará a detalle la metodología con sus fases y ventajas que se usará para el desarrollo del proyecto y así mismo como las herramientas con sus ventajas y desventajas que servirán para llegar al objetivo final.

2.1. Metodología Scrum

Scrum es un marco de trabajo ágil a través del cual las personas pueden abordar problemas complejos adaptativos a la vez que se entregan productos de forma eficiente y creativa con el máximo valor. Así, Scrum es una metodología que ayuda a los equipos a colaborar y realizar un trabajo de alto impacto. La metodología Scrum proporciona un plan de valores, roles y pautas para ayudar a tu equipo a concentrarse en la iteración y la mejora continua en proyectos complejos. (Martins, 2023)

2.1.1. Fases procesos de Scrum

A continuación, se describen las etapas de Scrum. (Mancuzo, 2020)



Figura 1. Fases de Scrum

Las fases de la metodología Scrum se reparten en 5 pasos o etapas de implementación:

1. Inicio
2. Planificación y estimación
3. Implementación
4. Revisión y retrospectiva
5. Lanzamiento

Fase de Inicio

La primera fase se encarga de estudiar y analizar el proyecto identificando las necesidades básicas del sprint.

Las preguntas a hacer en la fase de inicio son:

- ¿Qué quiero?
- ¿Cómo lo quiero?
- ¿Cuándo lo quiero?

La metodología Scrum da preferencia a la formación de equipos pequeños de mínimo 3 y máximo 5 personas, pues se facilita la fluidez de las ideas y se aporta creatividad al grupo.

Entre los primeros pasos de Scrum, tenemos 6 procesos:

- Crear la visión del proyecto
- Identificar al Scrum Master
- Formación del equipo Scrum
- Desarrollo de épicas
- Creación de la lista priorizada de pendientes del producto
- Realizar pan de lanzamiento

Fase de planificación y estimación

La segunda fase de Scrum incluye normalmente los siguientes pasos:

- Crear, estimar y comprometer historias de usuario.
- Identificar y estimar tareas.
- Crear el sprint backlog o iteración de tareas.

La clave para llevar una buena administración de los proyectos es hacer una planificación y estimación del sprint, lo que te ayudará a establecer metas fijas y a cumplir con los plazos.

Fase de Implementación

Al llegar a la tercera de las 5 fases de Scrum, nos topamos con la implementación del proyecto.

Es decir, la sala de reuniones donde se discute el sprint y se explora cómo optimizar el trabajo de cada grupo Scrum para darle forma definitiva al proyecto.

En la implementación se cumple con los siguientes procesos:

- Crear entregables.
- Realizar daily stand-up.
- Refinanciamiento del backlog priorizado del producto.

En la fase de implementación o desarrollo no deberían hacerse cambios innecesarios de última hora (se supone que para evitarlo existe una fase de planificación).

Fase de revisión y retrospectiva

Una vez que ya todo está maquetado e implementado, deberás hacer la revisión del proceso, que no es más que la autocrítica o evaluación interna del grupo respecto a su propio trabajo.

Es importante sumar opiniones constructivas y aportar soluciones viables.

Entre los pasos más importantes para realizar en esta fase tenemos:

- Demostrar y validar el sprint.
- Retrospectiva del sprint.

Fase de Lanzamiento

La última de las fases del método Scrum es el lanzamiento.

Con esto nos referimos al desenlace del proyecto y entrega del producto, donde deberías cumplir con 2 únicas tareas que son:

- Enviar entregables.
- Enviar retrospectiva del proyecto.

2.1.2. Ventajas y desventajas

Scrum puede ayudar a organizar el trabajo cuando se gestiona un proyecto o el cumplimiento de fases, es por ello que a continuación se presentan las ventajas y desventajas de Scrum. (Mancuzo, blog.comparasoftware, 2020)

Ventajas:

- Ayuda a establecer metas cuantificables y, así, mantener al equipo ocupado todo el tiempo para garantizar la productividad.
- La flexibilidad del método Scrum te permite modificar el hilo conductor de los hechos en cualquier momento, lo que favorece la resolución de conflictos u obstáculos sobre la marcha.
- Muestra el progreso del sprint o iteración de requerimientos.
- Evita el perfeccionismo innecesario.
- Se obtienen resultados rápidos y periodos de prueba muy cortos. Así, un producto puede estar listo para venta al público en poco tiempo.
- El método ofrece una mirada cercana y transparente del proyecto a todas las partes interesadas. Esto aporta confianza a los clientes.
- Motiva al cierre de los sprints (mejor tasa de cumplimiento).
- Te muestra una visión completa del proyecto.
- Usando un Software para Scrum, resulta muy fácil priorizar constantemente y saber distinguir entre los pendientes que pueden esperar y las tareas cruciales.

Desventajas:

- Al momento de trabajar con Scrum, todo el equipo debe conocer a fondo sus principios y marco teórico. Tiene que haber conocimiento de los roles de Scrum o podría haber dispersión y choque de funciones.

- En gran parte, el éxito de un proyecto conducido con el método Scrum dependerá de la preparación y experiencia del Scrum Master (más que de la innovación, creatividad o calidad de los insumos).
- Si por algún motivo quedan tareas sin finalizar, el resto de pendientes comenzará a postergarse indefinidamente, pues a menudo hay una relación lógica y secuencial entre las actividades del tablero.
- Puede ser difícil el manejo de Scrum en proyectos complejos.

2.2. Herramientas

Para el desarrollo de este proyecto se utilizará como herramientas: Visual Studio Code, Postman, PhpMyAdmin, Xampp, Composer, React, Node.js, GitHub, Google Chrome.

2.2.1. Visual Studio Code

Visual Studio Code es un editor de código en términos sencillos. Visual Studio Code es “un editor libre que ayuda al programador a escribir código, ayuda a depurar y corrige el código utilizando el método intelli-sense”. En términos normales, facilita a los usuarios a escribir el código de una manera fácil. Mucha gente dice que es la mitad de un IDE y un editor, pero la decisión depende de los codificadores.

Ventajas:

- **Es gratuito y de código abierto:** Es un software gratuito y de código abierto, lo que significa que no tienes que pagar nada para descargarlo y usarlo. Además, puedes personalizarlo y hacer cambios en el código fuente según tus necesidades.
- **Es altamente personalizable:** Es altamente personalizable, lo que significa que puedes cambiar la apariencia y la funcionalidad de la herramienta según tus necesidades. Puedes agregar extensiones, temas y atajos de teclado personalizados para hacer tu trabajo más eficiente.
- **Es compatible con varios lenguajes de programación:** Es compatible con una amplia gama de lenguajes de programación, incluyendo JavaScript, HTML, CSS, Python, Ruby y muchos más. Esto significa que no tienes que cambiar de herramienta cada vez que trabajas con un lenguaje diferente.
- **Tiene una gran comunidad:** Tiene una gran comunidad de usuarios y desarrolladores que contribuyen con extensiones y temas. Puedes encontrar una gran cantidad de recursos en línea para ayudarte a resolver problemas y aprender más sobre la herramienta.
- **Es compatible con varias plataformas:** Está disponible para Windows, macOS y Linux, lo que significa que puedes usarlo en la mayoría de las plataformas.

Desventajas:

- **Puede ser lento en proyectos grandes:** Puede ser lento cuando trabajas en proyectos grandes o con archivos grandes. Esto puede afectar tu productividad y hacer que tu trabajo sea menos eficiente.
- **Requiere conocimientos técnicos para personalizar:** Es altamente personalizable, puede requerir conocimientos técnicos para hacer cambios en el código fuente o crear extensiones personalizadas. Esto puede ser un problema para aquellos que no tienen experiencia en programación.
- **La interfaz de usuario puede ser confusa:** La interfaz de usuario de VS Code puede ser confusa para los nuevos usuarios, especialmente si vienen de otras herramientas de programación. Puede tomar tiempo acostumbrarse a la herramienta y descubrir dónde se encuentran todas las funciones.
- **Puede tener problemas de estabilidad:** Aunque VS Code es generalmente estable, puede haber problemas de estabilidad ocasionalmente. Esto puede afectar tu productividad y hacerte perder el trabajo que no has guardado.

2.2.2. Postman

Postman es una plataforma que permite y hace más sencilla la creación y el uso de APIs. Esta herramienta es muy útil para programar porque da la posibilidad hacer pruebas y comprobar el correcto funcionamiento de los proyectos que realizan los desarrolladores web. (Juan, 2022)

Ventajas:

- Postman tiene una comunidad enorme de usuarios.
- Su interfaz es muy intuitiva y sencilla.
- La plataforma puede integrarse con otras herramientas.
- Ofrece la posibilidad de añadir scripts (JavaScript) para automatizar, configurar pruebas o agregar validaciones.
- La herramienta permite realizar trabajos en los que se puede colaborar con el resto de miembros de un equipo.
- Y como hemos mencionado anteriormente, permite trabajar con colecciones.

Desventajas:

- Limitado para Pruebas Complejas: Aunque es excelente para pruebas de API básicas, puede ser menos adecuado para pruebas de carga y escenarios de prueba más complejos.

- Dependencia de la Conectividad a Internet: Algunas características avanzadas, como la sincronización de colecciones, pueden depender de la conectividad a Internet.
- Licencia para Características Avanzadas: Algunas características avanzadas, como la colaboración en equipo, pueden requerir una suscripción paga.

2.2.3. PhpMyAdmin

phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet, está disponible bajo la licencia GPL (General Public License).

Con esta herramienta puedes crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos. (Perez Hernandez, 2017)

Ventajas:

- Posee una interfaz web intuitiva
- Es desarrollada en php
- Sirve de referencia para la creación de phpgadmin
- Se encuentra bajo licencia gnu gpl que nos permite la libre distribución, modificación y uso.
- Se pueden importar datos de archivos cvs y sql.

Desventajas:

- Su código se basa en php.
- Limita la cantidad que maneja de bases de datos
- Solo administra bases de datos en mysql
- En cuestión de procedimientos de almacenar.

2.2.4. Xampp

Xampp es un servidor web local multiplataforma que permite la creación y prueba de páginas web u otros elementos de programación. Sin embargo, Xampp integra una serie de herramientas que potencian y facilitan la experiencia al desarrollador.

Es decir, Xampp en sí mismo no es un programa, sino un paquete de programas o software que contiene herramientas de gestión de base de datos. (Jesús, 2022)

Ventajas:

- **Fácil de instalar:** XAMPP es fácil de instalar en cualquier sistema operativo. La instalación es sencilla e incluye todo lo necesario para configurar un servidor web en tu computadora.

- **Completamente gratuito:** XAMPP es un software de código abierto y completamente gratuito. No tienes que pagar nada para usarlo.
- **Multiplataforma:** XAMPP es compatible con varios sistemas operativos, incluyendo Windows, Linux y MacOS. Puedes usarlo en cualquier sistema operativo que desees.
- **Incluye todo lo necesario:** XAMPP incluye todos los componentes necesarios para configurar y ejecutar un servidor web. No tienes que preocuparte por descargar e instalar cada componente por separado.
- **Fácil de usar:** XAMPP es fácil de usar, incluso para usuarios principiantes. Tiene una interfaz gráfica de usuario fácil de entender que te permite configurar y administrar tu servidor web.

Desventajas:

- **No es adecuado para entornos de producción:** XAMPP está diseñado para entornos de desarrollo y no para entornos de producción. No es recomendable usar XAMPP en un entorno de producción, ya que no es seguro.
- **No está diseñado para la seguridad:** XAMPP no está diseñado para la seguridad. Si usas XAMPP en un entorno de producción, debes asegurarte de tomar medidas adicionales para proteger tu servidor web.
- **No se actualiza con frecuencia:** XAMPP no se actualiza con frecuencia. Si usas XAMPP, debes asegurarte de mantenerlo actualizado para garantizar la seguridad y la compatibilidad con las últimas versiones de software.
- **Configuración manual:** XAMPP requiere cierta configuración manual para personalizar el servidor web según tus necesidades. Si no tienes experiencia en la configuración de servidores web, puede ser un poco complicado para ti.

2.2.5. Composer

Composer es un gestor de dependencias para el lenguaje PHP. Es el equivalente a npm para NodeJS o Maven para Java.

Como otros gestores de dependencias, nos sirve para instalar de una manera cómoda toda la serie de librerías y frameworks sobre las cuales construyamos cualquier tipo de proyecto PHP. Pero no solo eso, además de instalar Composer nos permite de una manera sencilla y cómoda mantener actualizadas las dependencias, con lo que nos asegura un óptimo estado del proyecto con menos esfuerzo.

Ventajas:

- **Gestión de Dependencias:** Composer simplifica la gestión de dependencias en proyectos PHP, permitiendo a los desarrolladores definir y declarar las bibliotecas y paquetes que necesita su aplicación.

- **Autocarga de Clases PSR-4:** Facilita la autocarga de clases según el estándar PSR-4, eliminando la necesidad de incluir manualmente archivos y mejorando la estructura y organización del código.
- **Versionamiento Semántico:** Utiliza versionamiento semántico para definir restricciones de versiones, lo que asegura que las dependencias se mantengan actualizadas de manera controlada y evita conflictos.

Desventajas:

- **Curva de Aprendizaje Inicial:** Para los desarrolladores nuevos en PHP o en el desarrollo moderno con Composer, puede haber una curva de aprendizaje al comprender la estructura del archivo composer.json y los comandos de Composer.
- **Conflicto de Dependencias:** En casos raros, puede surgir un conflicto de dependencias, especialmente al manejar proyectos grandes con múltiples dependencias, lo que podría requerir tiempo para resolver.

2.2.6. React

React es una biblioteca JavaScript de código abierto creada por el equipo de la compañía Facebook (Meta), junto a una gran comunidad de desarrolladores independientes. Se ha convertido en una de las tecnologías Front End más usadas, ya que permite construir interfaces de usuario dinámicas y escalables. Se usa en el desarrollo de aplicaciones web móviles, de una sola página o del servidor. (ebac, 2023)

Ventajas:

- Los componentes de React agilizan la creación de una interfaz sensible a cualquier cambio en un sitio web o una aplicación de cualquier complejidad.
- Gracias al DOM virtual, la biblioteca ahorra recursos y tráfico.
- El código de React tiene una lógica clara, es fácil de leer, entender y depurar, lo que ayuda a reducir errores.
- Las interfaces interactivas creadas con React garantizan una mejor experiencia de usuario.
- React es fácil de aprender, tiene una documentación accesible y muchos recursos gratuitos online.
- Dominar React es una de las habilidades más demandadas para conseguir el trabajo de desarrollo Front End.

Desventajas:

- Necesitas un conocimiento sólido de HTML y JavaScript para aprender la sintaxis de JSX.
- La biblioteca puede aumentar el tamaño de tu aplicación.
- React solo visualiza la interfaz, pero para crear un proyecto completo, necesitas una pila de tecnología.

2.2.7. Node.js

Node.js es un entorno de tiempo de ejecución de código abierto y multiplataforma que permite ejecutar código JavaScript fuera de un navegador web. Es decir, Node.js permite ejecutar JavaScript en el servidor. Fue creado por Ryan Dahl en 2009 y se basa en el motor JavaScript V8 de Google Chrome. (solbyte, 2023)

Ventajas:

- **Eficiencia y escalabilidad:** Utiliza un modelo de operaciones de entrada/salida sin bloqueo y basado en eventos, lo que permite manejar una alta concurrencia y un gran número de solicitudes simultáneas de manera eficiente. Esto lo hace ideal para aplicaciones web en tiempo real y de alto rendimiento.
- **Unificación del lenguaje:** Tanto en el servidor como en el cliente (con herramientas como React o Angular), se puede escribir todo el código en JavaScript. Esto facilita la comunicación y el intercambio de código entre el frontend y el backend, lo que puede mejorar la productividad del equipo de desarrollo.
- **Amplio ecosistema de módulos:** Cuenta con un gran repositorio de paquetes y módulos disponibles a través de npm. Esto facilita la reutilización de código y acelera el desarrollo de aplicaciones, ya que no es necesario crear todo desde cero.
- **Comunidad activa:** Hay una gran cantidad de recursos, documentación y soporte disponibles en línea. Además, hay una amplia variedad de herramientas y frameworks construidos sobre Node.js que pueden facilitar el desarrollo de aplicaciones web.

Desventajas:

- **Uso intensivo de CPU:** Puede no ser la mejor opción si tu aplicación realiza operaciones intensivas en el uso de la CPU, como procesamiento de imágenes o cálculos complejos. Dado que Node.js utiliza un solo hilo de ejecución, estas operaciones pueden bloquear el rendimiento general de la aplicación.
- **Madurez de algunas bibliotecas:** Aunque tiene un ecosistema de módulos y paquetes muy amplio, algunas bibliotecas pueden ser menos maduras o estables en comparación con otros lenguajes o frameworks más establecidos. Es importante investigar y evaluar las bibliotecas antes de utilizarlas en un proyecto.
- **Curva de aprendizaje:** Si el equipo de desarrollo no está familiarizado con JavaScript o con el paradigma de programación asíncrona, puede haber una curva de aprendizaje para comprender y aprovechar completamente las características de Node.js.

2.2.8. GitHub

GitHub es una plataforma de desarrollo colaborativo que permite a los equipos de programadores trabajar de manera conjunta en proyectos de software. Actúa como un repositorio en la nube donde se almacena el código fuente, y ofrece herramientas y funcionalidades para gestionar el ciclo de vida del desarrollo. Los desarrolladores pueden crear, clonar y colaborar en repositorios, realizar seguimiento de problemas, gestionar ramas de código, realizar revisiones de código, automatizar pruebas y despliegues, y mucho más. (Lizarzaburu Pezua, 2023)

Ventajas:

- **Colaboración efectiva:** GitHub permite una colaboración fluida y efectiva entre desarrolladores, lo cual facilita el trabajo en equipo y la revisión de código. Esto es especialmente útil en proyectos de código abierto donde múltiples colaboradores pueden contribuir y mejorar el código de manera conjunta.
- **Control de versiones:** Con el sistema de control de versiones de Git, en el que se basa GitHub, los desarrolladores pueden realizar un seguimiento exhaustivo de los cambios realizados en el código. Esto facilita la gestión de versiones, la identificación de errores y la reversión a versiones anteriores si es necesario.
- **Comunidad activa:** GitHub cuenta con una gran comunidad de desarrolladores que comparten proyectos de código abierto. Esto brinda la oportunidad de aprender de otros, colaborar en proyectos emocionantes y obtener retroalimentación valiosa de expertos en la materia.

Desventajas:

- **Curva de aprendizaje:** GitHub puede tener una curva de aprendizaje pronunciada para aquellos que no están familiarizados con el control de versiones y las herramientas asociadas. La comprensión y el dominio de los conceptos básicos de Git pueden llevar tiempo y esfuerzo.
- **Dependencia de la conectividad a Internet:** Para aprovechar todas las funciones de GitHub, se requiere una conexión a Internet estable. Esto puede ser una limitación en entornos donde la conectividad es deficiente o inexistente, lo que dificulta el acceso y la colaboración en los proyectos.
- **Privacidad y seguridad:** Aunque GitHub ofrece opciones de seguridad y privacidad, los proyectos alojados en la plataforma son accesibles de forma pública por defecto. Esto puede plantear preocupaciones de seguridad si se manejan datos sensibles o si se requiere un mayor control de acceso.

2.2.9. Laravel

Laravel es un framework de desarrollo web de código abierto y basado en PHP que sigue el patrón de diseño MVC (Modelo-Vista-Controlador). Creado por Taylor Otwell, Laravel ha ganado popularidad debido a su sintaxis elegante, su enfoque en la legibilidad del código y su conjunto robusto de características que facilitan el desarrollo rápido y eficiente de aplicaciones web.

Ventajas:

- **Elegancia y Claridad de Código:** Laravel fomenta una sintaxis elegante y expresiva, lo que facilita la lectura y escritura de código, y promueve las mejores prácticas de desarrollo.
- **Arquitectura MVC:** Utiliza el patrón de diseño Modelo-Vista-Controlador para organizar la lógica de la aplicación, facilitando la separación de preocupaciones y la mantenibilidad del código.
- **ORM Eloquent:** Incluye Eloquent, un ORM (Object-Relational Mapping) que simplifica las operaciones de base de datos, permitiendo a los desarrolladores interactuar con la base de datos utilizando objetos en lugar de SQL directo.
- **Blade Templating:** Blade es el motor de plantillas de Laravel que permite la creación de vistas de manera eficiente y la reutilización de componentes, mejorando la legibilidad del código HTML/PHP.

Desventajas:

- **Curva de Aprendizaje Inicial:** Para los principiantes en PHP o en el desarrollo web, Laravel puede tener una curva de aprendizaje inicial, especialmente si no están familiarizados con el concepto de frameworks MVC.
- **Actualizaciones Frecuentes:** Las actualizaciones regulares son beneficiosas para la seguridad y la mejora continua, pero podrían requerir ajustes en el código existente al actualizar a nuevas versiones.
- **Rendimiento en Aplicaciones Muy Grandes:** En aplicaciones extremadamente grandes, algunos desarrolladores pueden encontrar que ciertos componentes de Laravel, como Eloquent, pueden tener un impacto en el rendimiento.

2.2.10. Google Chrome

Google Chrome es un navegador web de código cerrado desarrollado por Google, aunque derivado de proyectos de código abierto. Está disponible gratuitamente. El nombre del navegador deriva del término en inglés usado para el marco de la interfaz gráfica de usuario.

Ventajas:

- Administrador de tareas.
- Gran variedad de extensiones para instalar.

- Navegación de incógnito.
- Integración con más servicios de Google.
- Rapidez.
- Seguridad.

Desventajas:

- Alto consumo de la memoria del ordenador
- Privacidad de los datos
- Poco soporte técnico

3 Resultados

En este capítulo de acuerdo con la metodología planteada en el capítulo 2, se describirán los resultados obtenidos.

3.1 Iniciación

3.1.1 Visión del Proyecto

La visión del proyecto es que los encargados del laboratorio puedan visualizar un listado de las muestras, análisis y ordenes de trabajo que se realizan dentro de dicho lugar. Así mismo los empleados del laboratorio llevaran un mejor control de los datos que manejan ya que todos los datos estarán guardados dentro del sistema.

3.1.2 Formación del equipo Scrum

El equipo Scrum constara de los siguientes roles:

Equipo Scrum	
Rol	Nombre
Scrum Master	David Ángel Hernández Sánchez
Prduct Owner	David Ángel Hernández Sánchez
Development team (Backend)	Javier García Maldonado
Development team (Frontend)	Samuel Martínez Arenas

Tabla 1. Formación del equipo scrum

3.2 Planificación y estimación

Para el desarrollo de este sistema las entregas serán las siguientes:

Baclog		
Numero de tarea	Tarea	Total, de horas
Sprint 1		
1	Instalación de la plantilla	50
2	Diseñar la interfaz para poder iniciar sesión	
3	Hacer conexión a la base de datos para obtener usuario y contraseña	
4	Realizar pruebas	
Sprint 2		
1	Diseñar catálogos de usuarios, roles y análisis	100
2	Realizar pruebas	

3	Hacer las correcciones necesarias	
Sprint 3		
1	Diseñar el catálogo de clientes	150
2	Realizar pruebas para revisar la funcionalidad correcta	
3	Hacer correcciones	
Sprint 4		
1	Diseño de la interfaz y consultar API de muestras	150
2	Consultar tabla de análisis para saber que análisis se realizara a la muestra que se entrego	
3	Realizar pruebas en la interfaz	
Sprint 5		
1	Diseño del catálogo ordenes de trabajo.	150
2	Consultar las tablas de muestras y clientes para poder hacer la relación.	
3	Consultar API.	
4	Realizar pruebas de funcionalidad.	

Tabla 2. Backlog

3.3 Implementación

Para este desarrollo del sistema web por parte del frontend se planea este periodo de lanzamiento de las entrega. Y se descubrirá como fue el desarrollo de cada uno de los Sprints que se mencionaron en el punto anterior.

3.3.1 Sprint 1

Para la planificación del sprint 1 se realizaron las siguientes tareas como se muestra en la siguiente tabla.

Sprint 1			
No. De Tarea	Duración del sprint	Fecha del Inicio de Sprint	Fecha fin del Sprint
1	1 semana	11/Septiembre/2023	15/Septiembre/2023
2			
3			
4			

Tabla 3. Sprint 1

Implementación del Sprint 1

Como primera instancia fue hacer la instalación de los paquetes en la plantilla.

```
npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @babel/plugin-proposal-optional-chaining@7.21.0: This proposal has been merged to the ECMAScript standard and thus this plugin is no longer maintained. Please use @babel/plugin-transform-optional-chaining instead.
npm WARN deprecated rollup-plugin-terser@5.3.1: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-terser
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated trim@0.0.1: Use String.prototype.trim() instead
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive security updates since 2019. Upgrade to chokidar 3 with 15 x fewer dependencies
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated sourcemap-codec@1.4.8: Please use @jridgewell/sourcemap-codec instead
npm WARN deprecated querystring@0.2.1: The querystring API is considered Legacy. new code should use the URLSearchParams API instead.
npm WARN deprecated w3c-hr-time@1.0.2: Use your platform's native performance.now() and performance.timeOrigin.
npm WARN deprecated sane@4.1.0: some dependency vulnerabilities fixed, support for node < 10 dropped, and newer ECMAScript syntax/features added
npm WARN deprecated rollup-plugin-babel@4.4.0: This package has been deprecated and is no longer maintained. Please use @rollup/plugin-babel.
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
npm WARN deprecated babel-eslint@10.1.0: babel-eslint is now @babel/eslint-parser. This package will no longer receive updates.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/joi@15.1.1: Switch to 'npm install joi'
##### \ reify:react-dom: http fetch GET 200 https://registry.npmjs.org/@babel/traverse/-/traverse-7.23.3.t
```

Figura 2. Instalación de paquetes en la plantilla

Diseño de la interfaz en el navegador

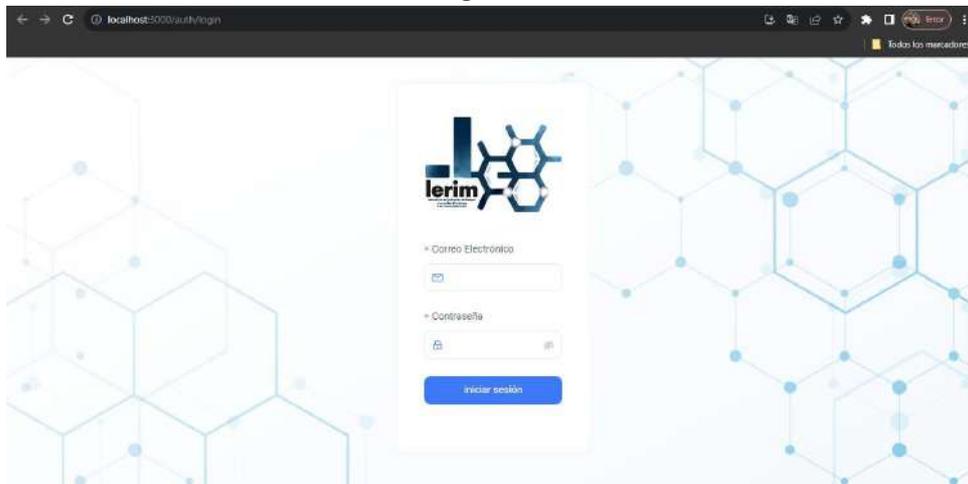


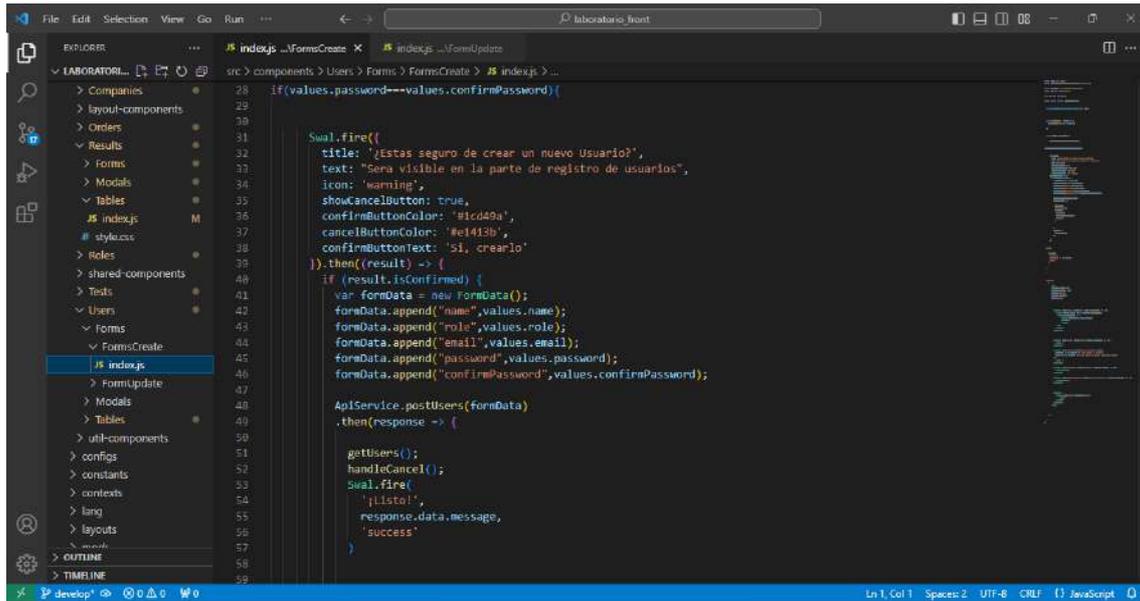
Figura 3. Interfaz de inicio de sesión

Implementación del Sprint 2

Usuarios

El catálogo de usuarios cuenta con las opciones de ver, crear y actualizar. Y a continuación se mostrarán imágenes de cada esas opciones.

- **Crear Usuario**



```
if(values.password===values.confirmPassword){  
  
Swal.fire({  
  title: '¿Estas seguro de crear un nuevo Usuario?',  
  text: "Sena visible en la parte de registro de usuarios",  
  icon: 'warning',  
  showCancelButton: true,  
  confirmButtonColor: '#1c409a',  
  cancelButtonColor: '#e1433b',  
  confirmButtonText: 'Si, crearlo'  
}).then((result) => {  
  if (result.isConfirmed) {  
    var formData = new FormData();  
    formData.append("name", values.name);  
    formData.append("role", values.role);  
    formData.append("email", values.email);  
    formData.append("password", values.password);  
    formData.append("confirmPassword", values.confirmPassword);  
  
    ApiService.postUsers(formData)  
      .then(response => {  
  
        getUsers();  
        handleCancel();  
        Swal.fire(  
          '¡Listo!',  
          response.data.message,  
          'success'  
        );  
      });  
  }  
});
```

Figura 6. Código para crear un nuevo Usuario.

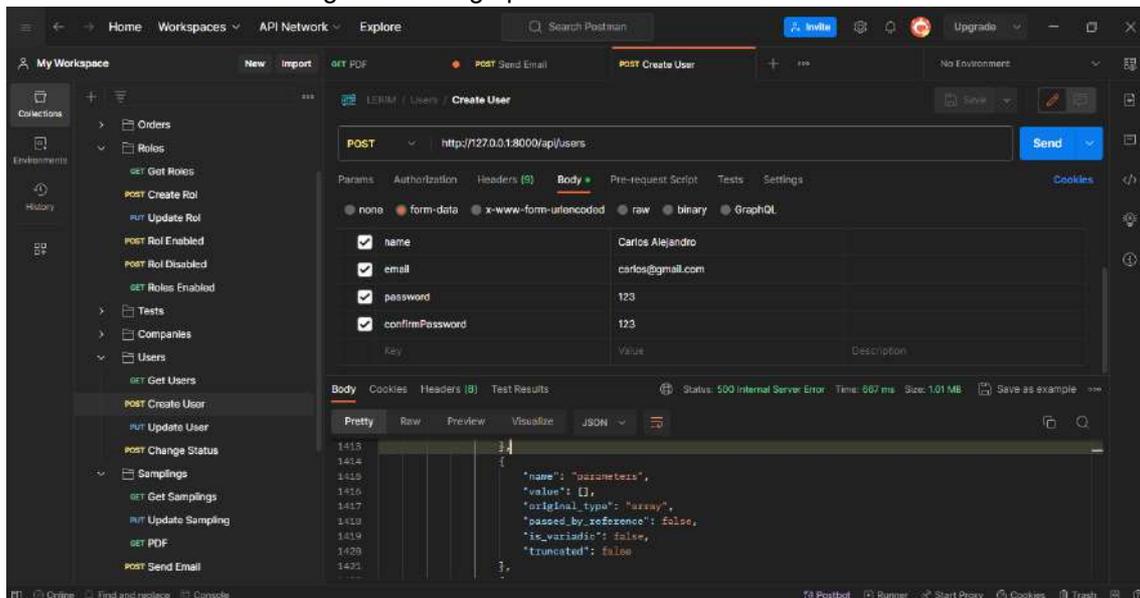


Figura 7. Uso de la API para crear un nuevo Usuario.

Nuevo Usuario [X]

* Rol

* Nombre

* Correo Electrónico

* Contraseña

* Confirmar Contraseña

Guardar

Figura 8. Interfaz para crear un Nuevo Usuario.

- **Actualizar Usuario**

```

if(showPasswordField){
  if(values.password===values.confirmPassword){

    Swal.fire({
      title: '¿Estas seguro de actualizar el Usuario?',
      text: 'Sera visible en la parte de registro de usuarios',
      icon: 'warning',
      showCancelButton: true,
      confirmButtonColor: '#1cd49a',
      cancelButtonColor: '#e1413b',
      confirmButtonText: 'Si, actualizar'
    }).then((result) => {
      if (result.isConfirmed) {
        var formData = new FormData();
        formData.append("id",values.id);
        formData.append("name",values.name);
        formData.append("role",values.role);
        formData.append("email",values.email);
        formData.append("password",values.password);
        formData.append("confirmPassword",values.confirmPassword);
        var from={id:values.id,
          name:values.name,
          role:values.role,
          email:values.email,
          password:values.password}

        ApiService.updateUsers(from)
        .then(response => {

```

Figura 9. Código para actualizar un Usuario.

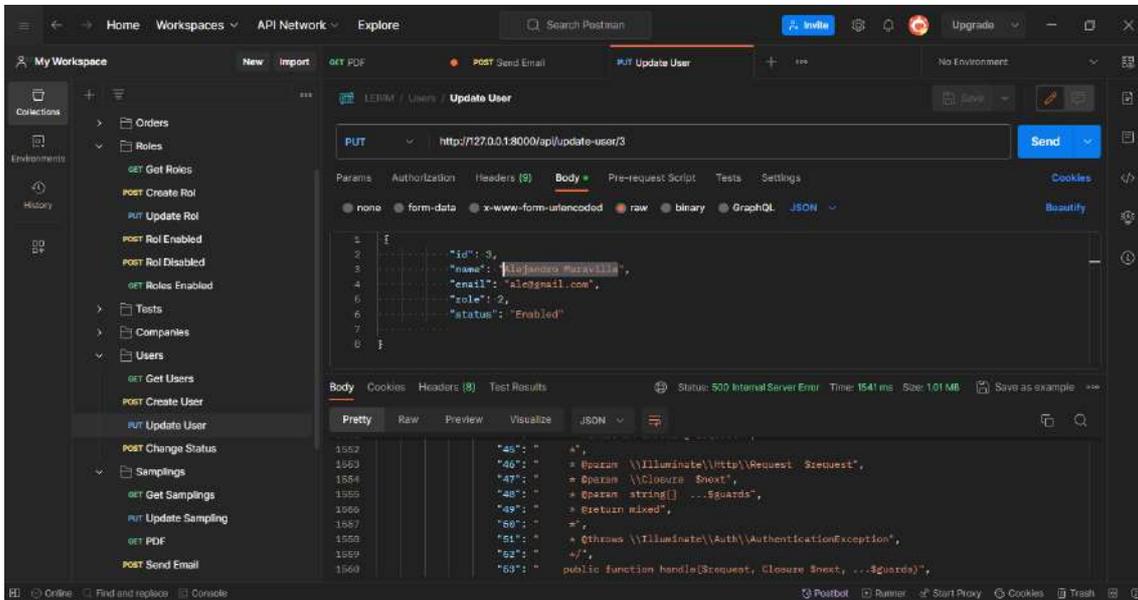


Figura 10. Uso de la API para actualizar un Usuario.

A screenshot of a web form titled 'Actualizar Usuario'. The form contains the following fields and elements:

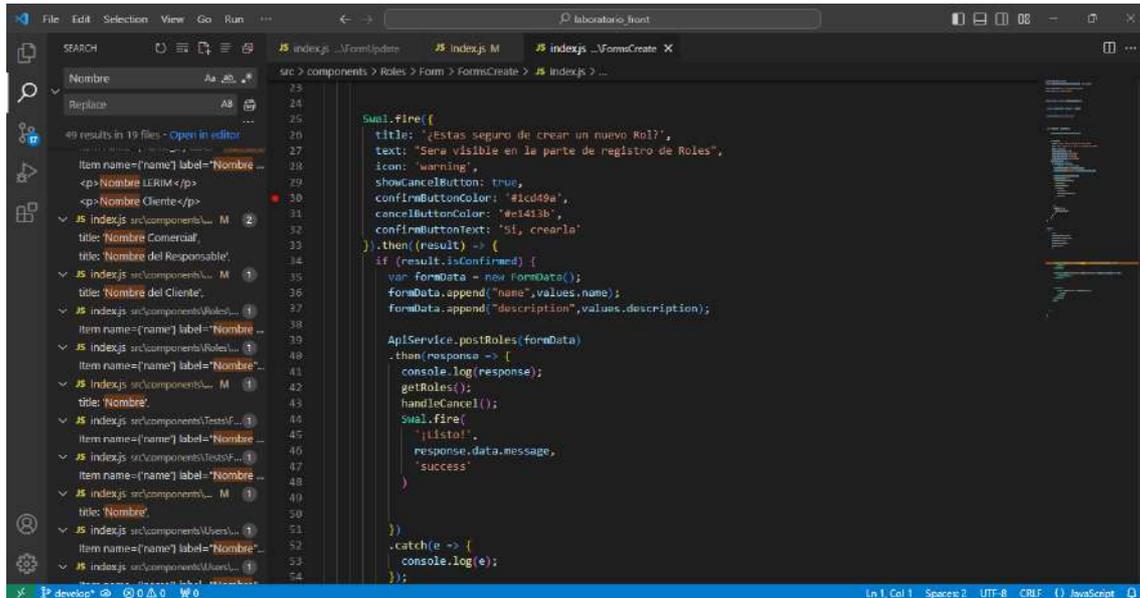
- A dropdown menu for 'Rol' with the value 'Administrador' selected.
- A text input field for 'Nombre' containing 'Usuario Administrador'.
- A text input field for 'Correo Electrónico' containing 'admin@gmail.com'.
- A checkbox labeled 'Cambiar Contraseña' which is currently unchecked.
- A blue button labeled 'Guardar' at the bottom left.

Figura 11. Interfaz para actualizar un Usuario.

Roles

El catálogo de usuarios cuenta con las opciones de ver, crear y actualizar. Y a continuación se mostrarán imágenes de cada esas opciones.

- **Crear Roles**



```
Swal.fire({
  title: '¿Estas seguro de crear un nuevo Rol?',
  text: 'Será visible en la parte de registro de Roles',
  icon: 'warning',
  showCancelButton: true,
  confirmButtonColor: '#1cc49a',
  cancelButtonColor: '#e1413b',
  confirmButtonText: 'SI, crearla'
}).then((result) => {
  if (result.isConfirmed) {
    var formData = new FormData();
    formData.append("name", values.name);
    formData.append("description", values.description);

    ApiService.postRoles(formData)
      .then(response => {
        console.log(response);
        getRoles();
        handleCancel();
        Swal.fire(
          '¡Listo!',
          response.data.message,
          'success'
        );
      })
      .catch(e => {
        console.log(e);
      });
  }
});
```

Figura 14. Código para crear un nuevo Rol.

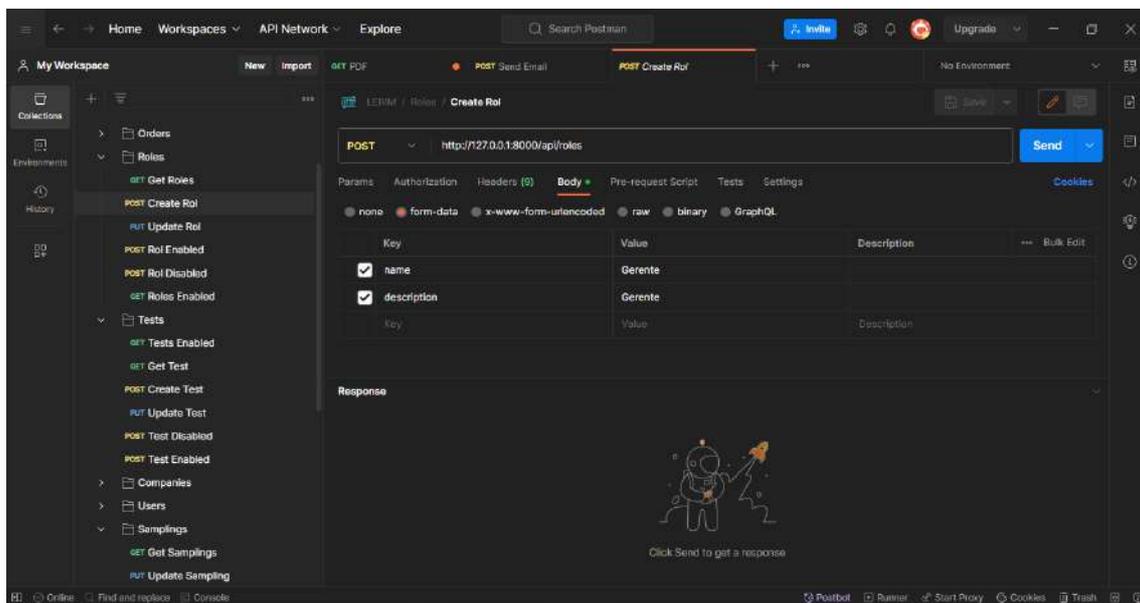


Figura 15. Uso de la API para crear un nuevo Rol.

Nuevo Usuario
✕

*** Rol**

*** Nombre**

*** Correo Electrónico**

*** Contraseña**

*** Confirmar Contraseña**

Figura 16. Interfaz para crear un nuevo Rol.

- **Actualizar Roles**

```

src > components > Roles > Form > FormUpdate > FormUpdate > onFinish
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
const { getRoles, handleCancel, dataRole } = props;
console.log(JSON.stringify(dataRole));
const onFinish = (values) => {
  Swal.fire({
    title: '¿Estás seguro de actualizar el Rol?',
    text: 'Será visible en la parte de registro de Roles',
    icon: 'warning',
    showCancelButton: true,
    confirmButtonColor: '#007bff',
    cancelButtonColor: '#dc3545',
    confirmButtonText: 'Sí, actualizarlo'
  }).then((result) => {
    if (result.isConfirmed) {
      ApiService.updateRol(values)
        .then(response => {
          console.log(response);
          getRoles();
          handleCancel();
        })
        .catch(e => {
          console.log(e);
        });
    }
  });
}

```

Figura 17. Código para actualizar un Rol.

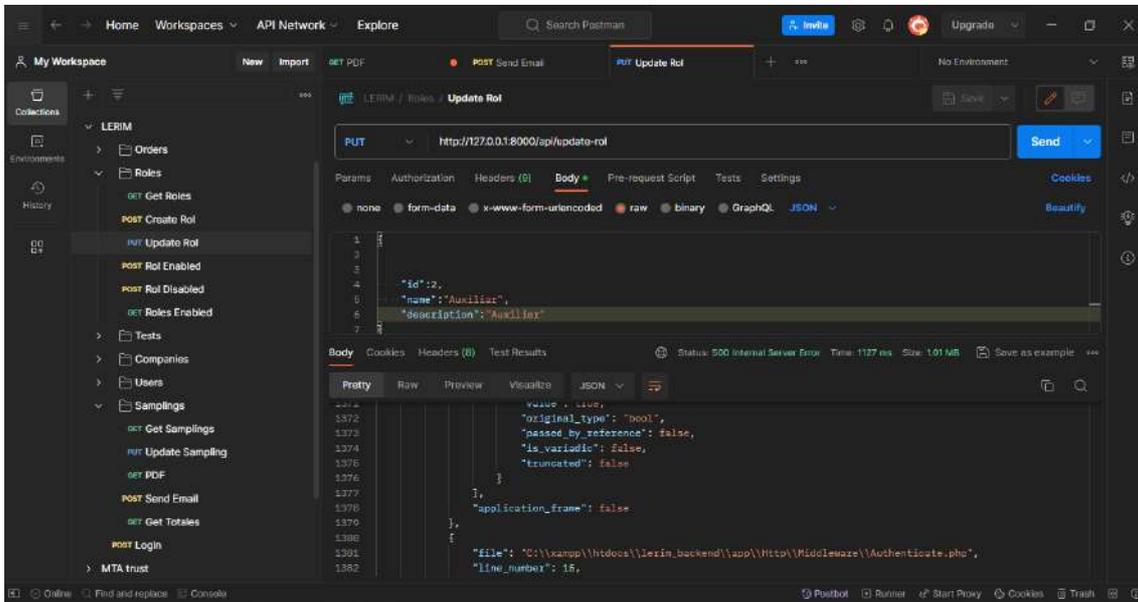


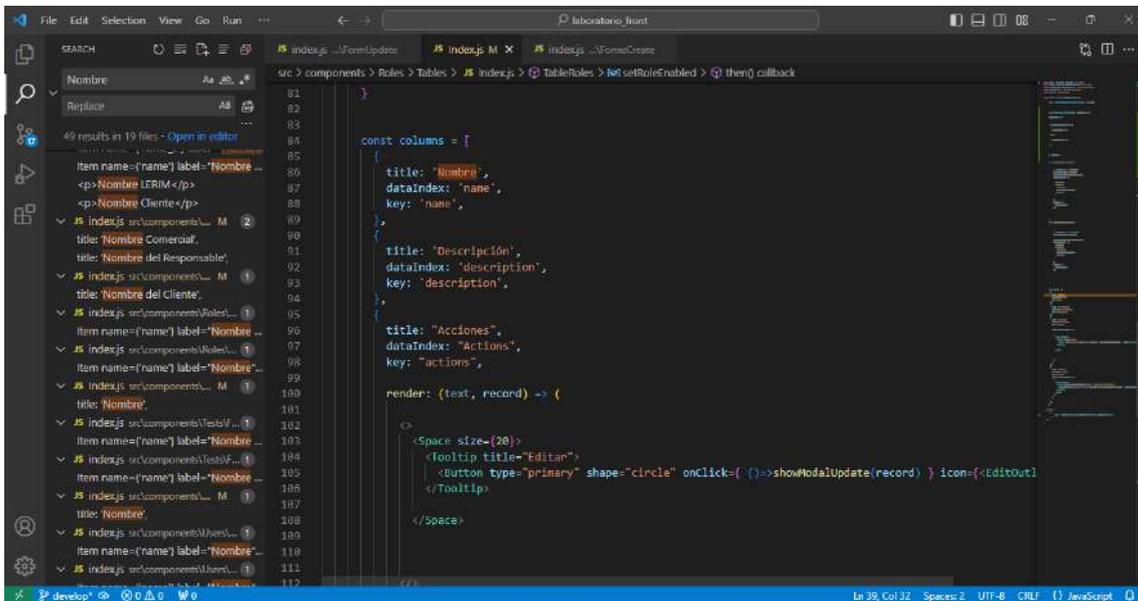
Figura 18. Uso de la API para actualizar un Rol.

The form is titled "Actualizar Usuario" and contains the following fields:

- * Rol**: A dropdown menu with "Administrador" selected.
- * Nombre**: A text input field containing "Usuario Administrador".
- * Correo Electrónico**: A text input field containing "admin@gmail.com".
- Cambiar Contraseña**: An unchecked checkbox.
- Guardar**: A blue button at the bottom.

Figura 19. Interfaz para actualizar un Rol.

- Ver Roles



The image shows a code editor window with a search sidebar on the left and a code editor on the right. The search sidebar shows 49 results in 19 files. The code editor shows the following code:

```
src > components > Roles > Tables > JS > index.js > TableRoles > @setRoleEnabled > then() callback  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112
```

```
const columns = [  
  {  
    title: 'Nombre',  
    dataIndex: 'name',  
    key: 'name',  
  },  
  {  
    title: 'Descripción',  
    dataIndex: 'description',  
    key: 'description',  
  },  
  {  
    title: 'Acciones',  
    dataIndex: 'actions',  
    key: 'actions',  
  },  
];  
  
render: (text, record) => (  
  <Space size={20}>  
    <Tooltip title="Editar">  
      <Button type="primary" shape="circle" onClick={()=>showModalUpdate(record)} icon={<EditOutlined/>} />  
    </Tooltip>  
  </Space>  
)
```

Figura 20. Código para ver a los roles registrados.

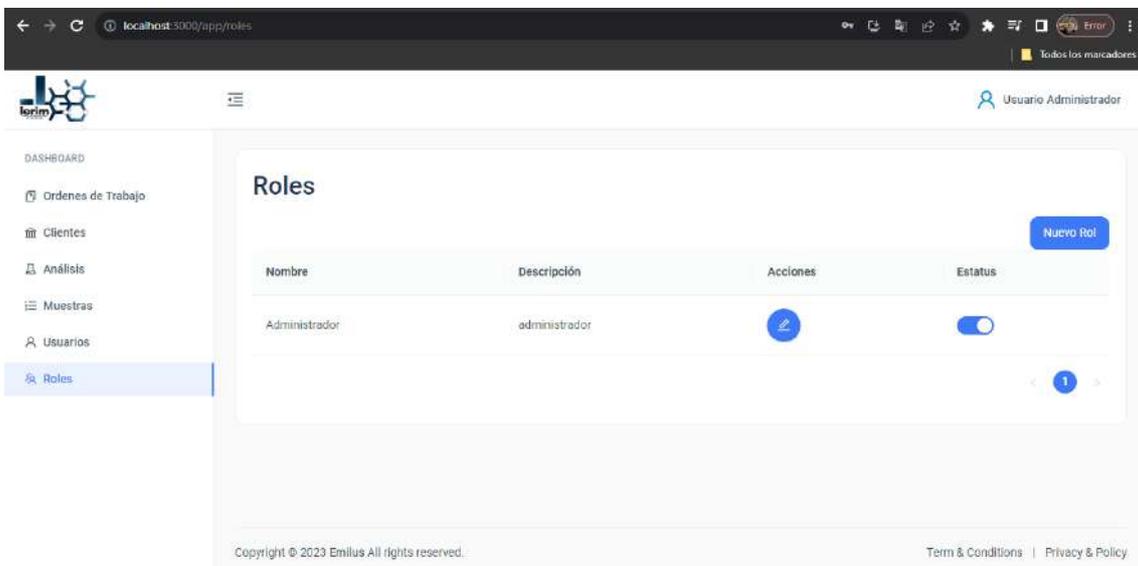
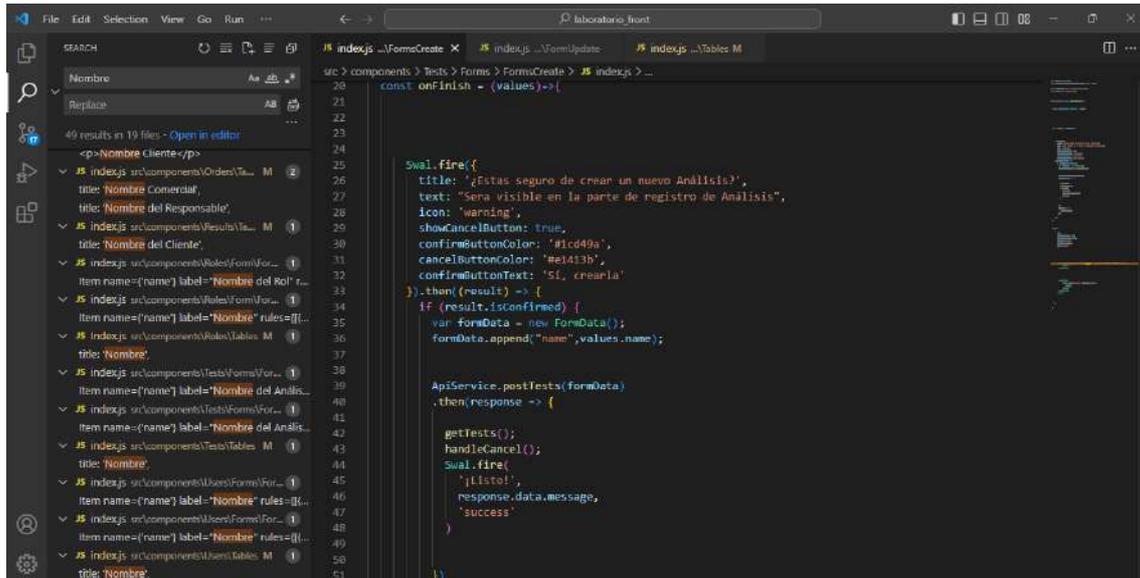


Figura 21. Interfaz para ver a los roles registrados.

Análisis

El catálogo de análisis cuenta con las opciones de ver, crear y actualizar. Y a continuación se mostrarán imágenes de cada esas opciones.

- **Crear Análisis**



```
src > components > Tests > Forms > FormCreate > JS index.js > ...
20 const onFinish = (values) => {
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Figura 22. Código para crear un nuevo Análisis.

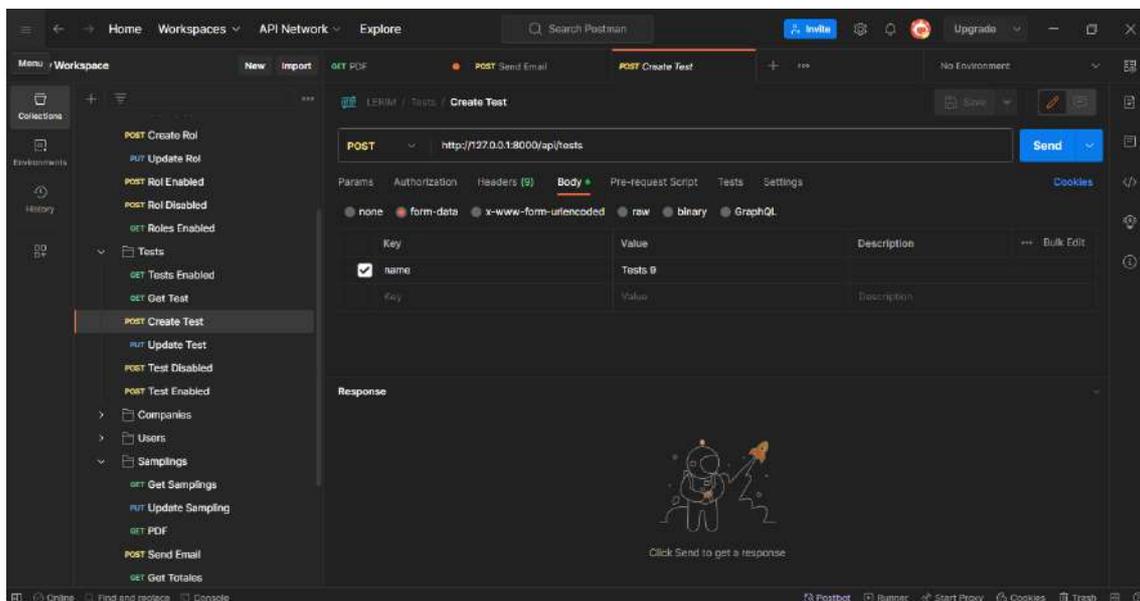


Figura 23. Uso de la API para crear un nuevo Análisis.



Figura 24. Interfaz para crear un nuevo Rol.

- **Actualizar Análisis**

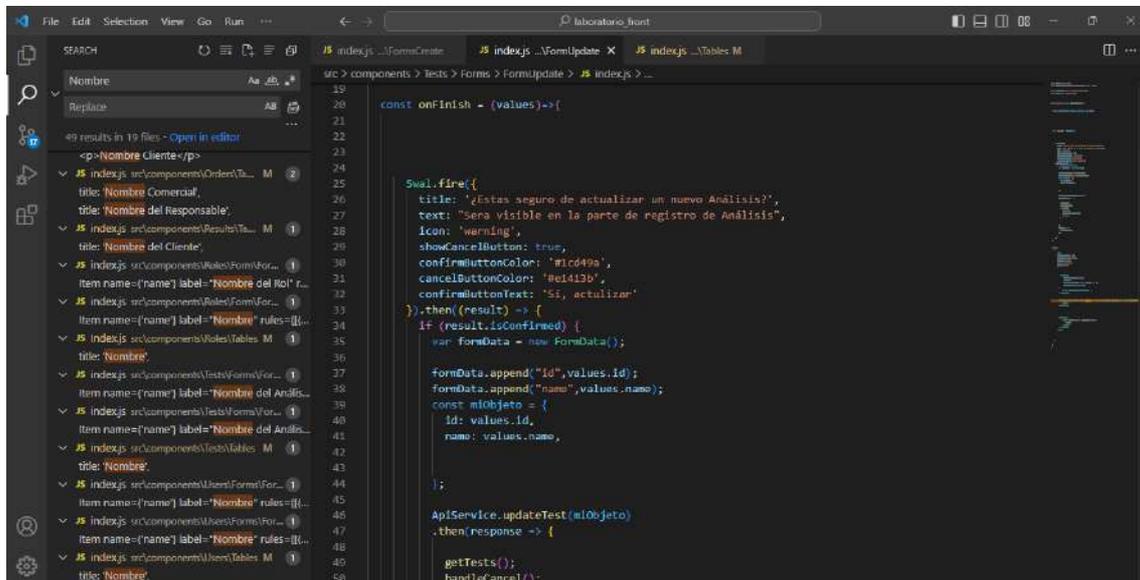


Figura 25. Código para actualizar un Análisis.

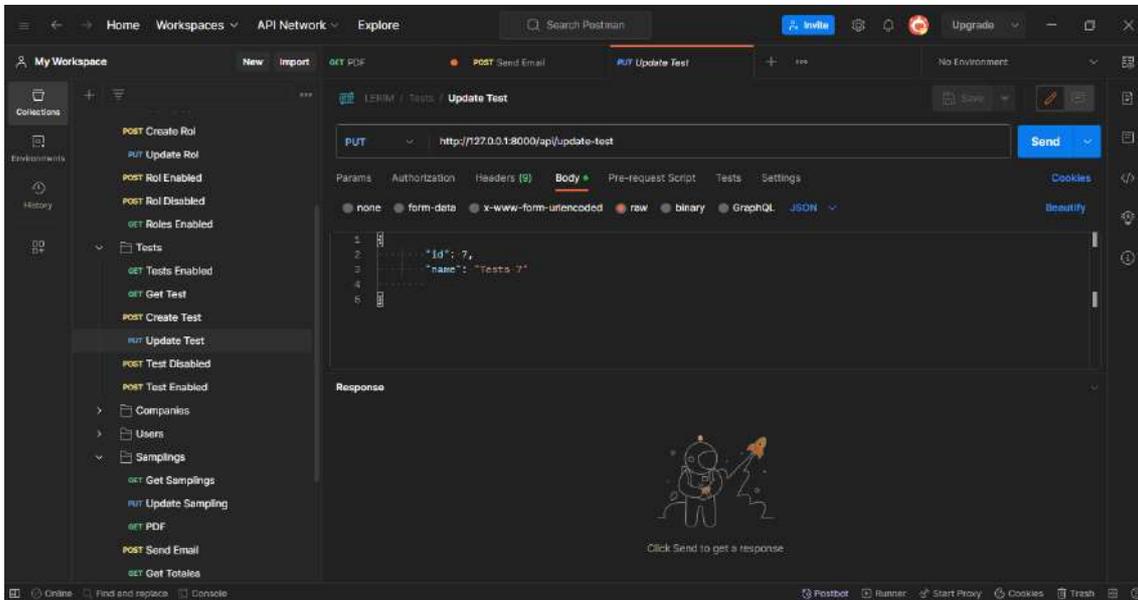


Figura 26. Uso de la API para actualizar un Análisis.

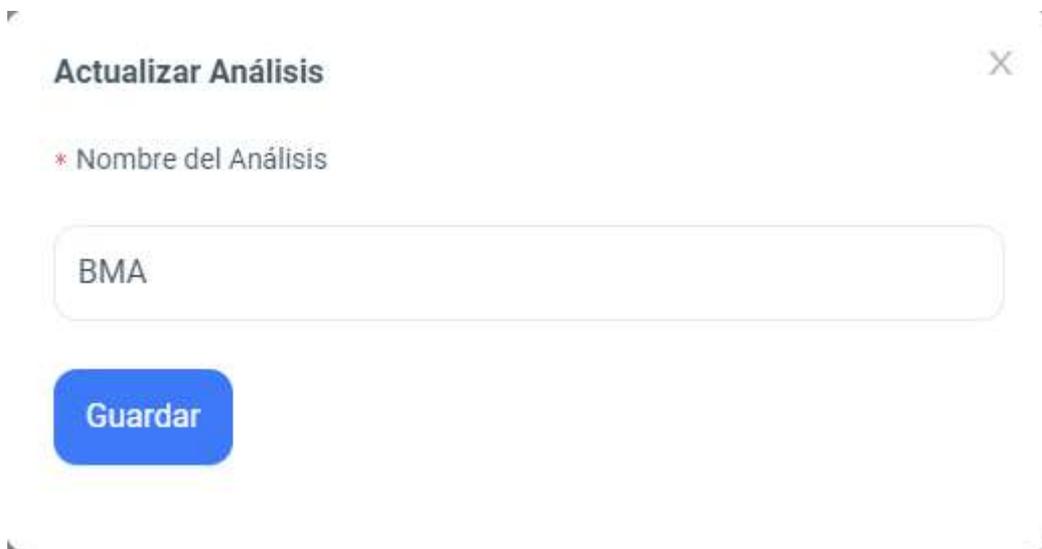


Figura 27. Interfaz para actualizar un Análisis.

- Ver Análisis

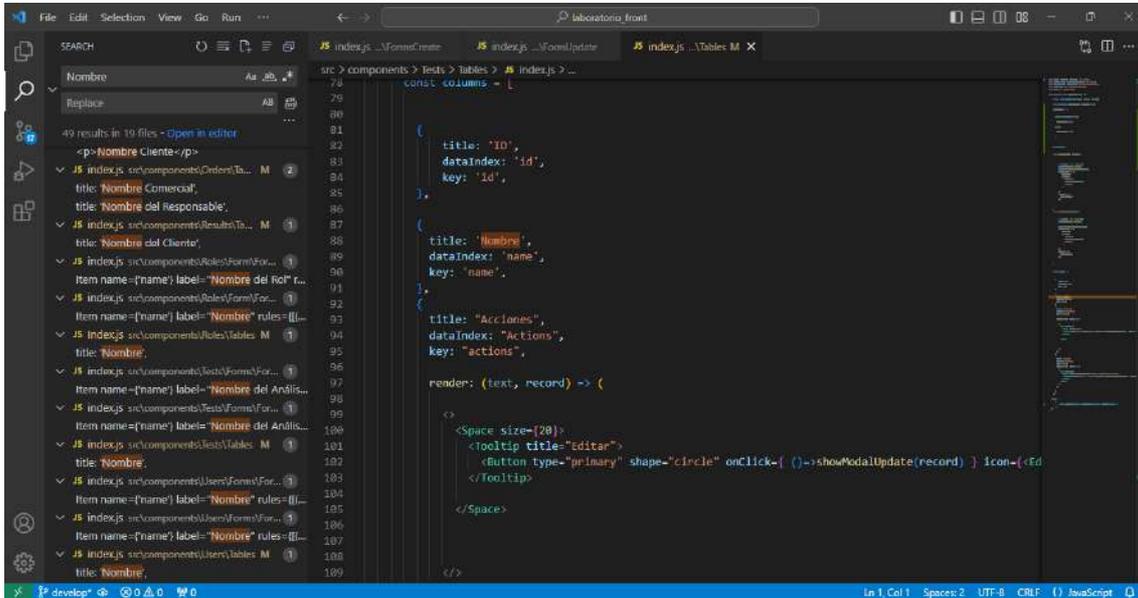


Figura 28. Código para ver a los análisis registrados.

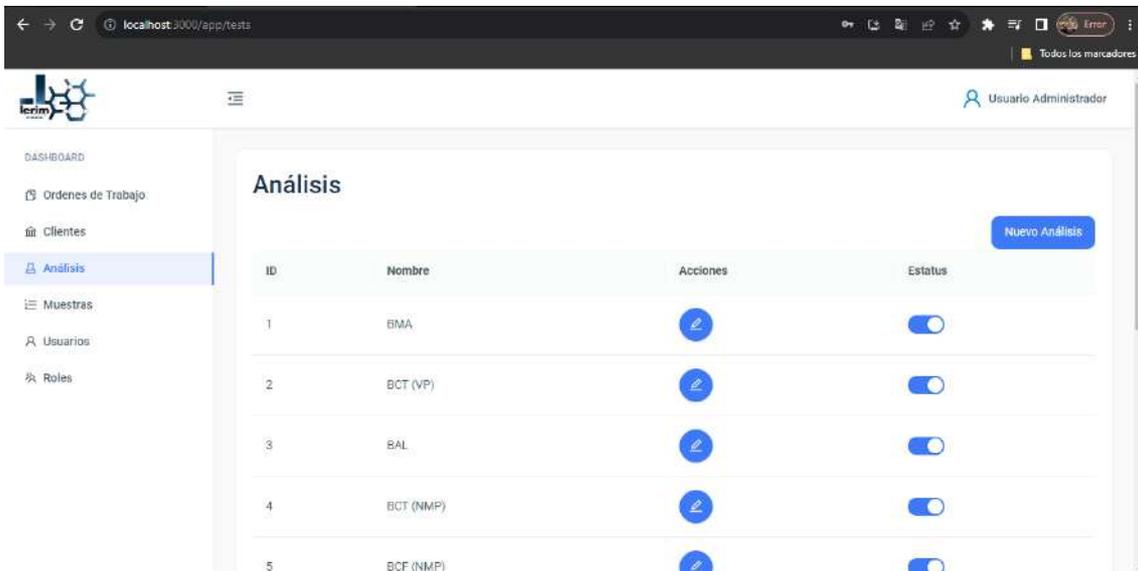


Figura 29. Interfaz para ver a los análisis registrados.

Sprint 3

Diseñar la interfaz para el catálogo de clientes con sus respectivas funcionalidades como agregar, modificar y activar o desactivar el status de cada uno de ellos. En la siguiente tabla se muestra el tiempo en que se realizaron las siguientes tareas.

Sprint 3			
No. De Tarea	Duración del sprint	Fecha del Inicio de Sprint	Fecha fin del Sprint
1	3 semanas	02/Octubre/2023	20/Octubre/2023
2			
3			

Tabla 5. Sprint 3

Implementación del Sprint 3

- Crear Clientes

The screenshot shows a code editor with the following HTML code for a form:

```

<Row justify="space-between">
  <Col xs=(24) sm=(11) md=(11) lg=(11)>
    <Form.Item name="name_responsable" label="Nombre del Responsable" rules={[{ required: true }]}>
      <Input />
    </Form.Item>
  </Col>
  <Col xs=(24) sm=(11) md=(11) lg=(11)>
    <Form.Item name="name" label="Nombre Comercial" rules={[{ required: true }]}>
      <Input />
    </Form.Item>
  </Col>
  <Col xs=(24) sm=(11) md=(11) lg=(11)>
    <Form.Item name="rfc" label="R.F.C" rules={[{ required: true }]}>
      <Input />
    </Form.Item>
  </Col>
  <Col xs=(24) sm=(11) md=(11) lg=(11)>
    <Form.Item name="email" label="Correo Electrónico">
      <Input />
    </Form.Item>
  </Col>
  <Col xs=(24) sm=(11) md=(11) lg=(11)>
    <Form.Item name="phoneNumber" label="Teléfono">
      <Input />
    </Form.Item>
  </Col>
</Row>

```

Figura 30. Código para crear un nuevo Cliente.

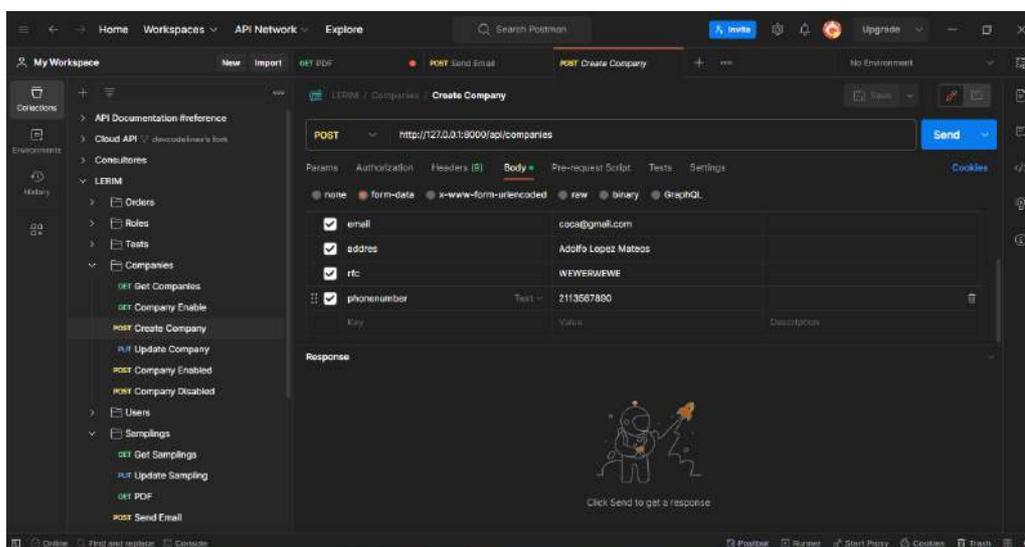


Figura 31. Uso de la API para crear un nuevo Análisis.

Nuevo Cliente
✕

*** Nombre del Responsable**

*** Nombre Comercial**

*** R.F.C**

Correo Electrónico

Teléfono

*** Dirección**

CDI

CP

Régimen

Figura 32. Interfaz para crear un nuevo Cliente.

- **Actualizar Clientes**

```

Swal.fire({
  title: '¿Estás seguro de actualizar el Cliente?',
  text: 'Será visible en la parte de registro de Clientes',
  icon: 'warning',
  showCancelButton: true,
  confirmButtonColor: '#1c499a',
  cancelButtonColor: '#e1413b',
  confirmButtonText: 'Sí, actualizar'
}), then((result) => {
  if (result.isConfirmed) {
    var formData = new FormData();
    formData.append("id", values.id);
    formData.append("name_responsible", values.name_responsible);
    formData.append("rfc", values.rfc);
    formData.append("name", values.name);
    formData.append("email", values.email);
    formData.append("phonenumber", values.phonenumber);
    formData.append("address", values.address);

    const miObjeto = {
      id: values.id,
      name_responsible: values.name_responsible,
      rfc: values.rfc,
      name: values.name,
      email: values.email,
      phonenumber: values.phonenumber,
      address: values.address,
      cdi: values.cdi,
      cp: values.cp,
      tax_regime: values.tax_regime,
    };
  }
});

```

Figura 33. Código para actualizar un Cliente.

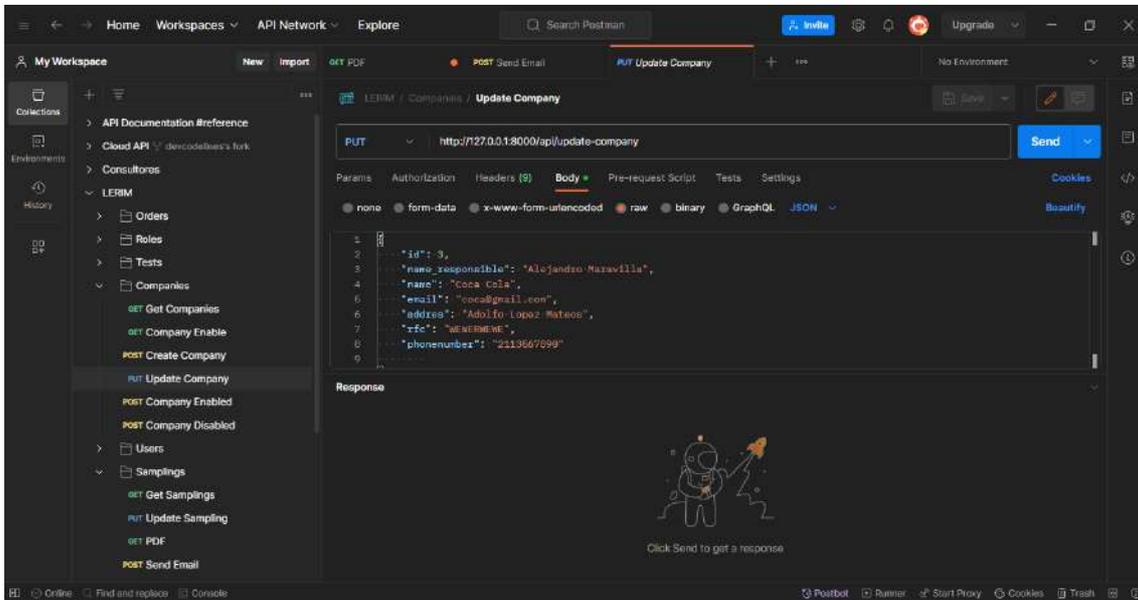


Figura 34. Uso de la API para actualizar un Cliente.

A web form titled 'Actualizar Cliente' with a close button in the top right corner. The form is organized into two columns. The left column contains: 'Nombre del Responsable' (required) with the value 'Carlos Maravilla'; 'R.F.C' (required) with the value '1234567889000'; 'Teléfono' with the value '221823456834'; 'CDI' (empty); and 'Régimen' (empty). The right column contains: 'Nombre Comercial' (required) with the value 'Agua Purificada'; 'Correo Electrónico' with the value 'carlos@gmail.com'; 'Dirección' (required) with the value 'Avenida Reforma 2202'; and 'CP' with the value '72700'. A blue 'Guardar' button is located at the bottom left of the form.

Figura 35. Interfaz para actualizar un Cliente.

- Ver Clientes

```

const columns = [
  {
    title: 'Nombre Comercial',
    dataIndex: 'name',
    key: 'name',
  },
  {
    title: 'Nombre del responsable',
    dataIndex: 'name_responsible',
    key: 'name_responsible',
  },
  {
    title: 'R.F.C.',
    dataIndex: 'rfc',
    key: 'rfc',
  },
  {
    title: 'Correo Electrónico',
    dataIndex: 'email',
    key: 'email',
  },
  {
    title: 'Teléfono',
    dataIndex: 'phonenumber',
    key: 'phonenumber',
  },
  {
    title: 'Dirección',
    dataIndex: 'address',
    key: 'address',
  }
];
  
```

Figura 36. Código para ver a los clientes registrados.

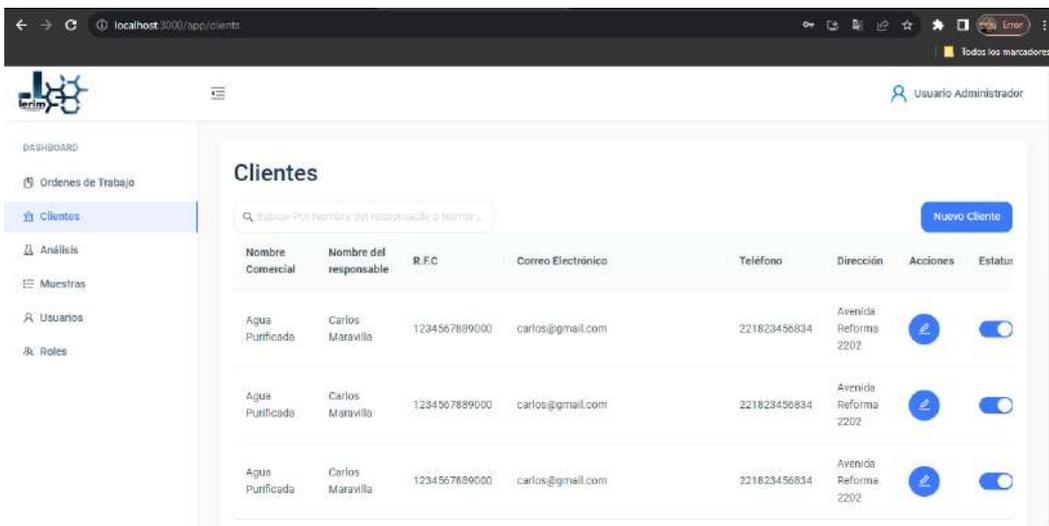


Figura 37. Interfaz para ver a los clientes registrados.

Sprint 4

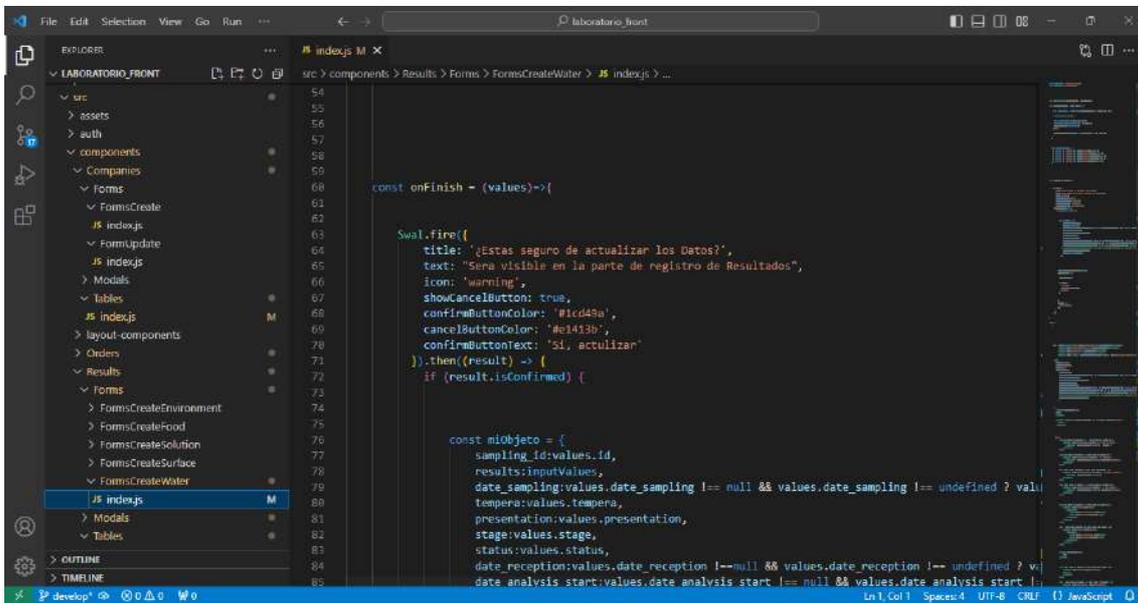
Diseñar la interfaz para el catálogo de muestras con sus respectivas funcionalidades como modificar y activar o desactivar el status de cada uno de ellos. En la siguiente tabla se muestra el tiempo en que se realizaron las siguientes tareas.

Sprint 4			
No. De Tarea	Duración del sprint	Fecha del Inicio de Sprint	Fecha fin del Sprint
1	3 semanas	23/Octubre/2023	10/Noviembre/2023
2			
3			
4			

Tabla 6. Sprint 4

Implementación del Sprint 4

- Actualizar Muestras



```
const onFinish = (values) => {  
  Swal.fire({  
    title: '¿Estas seguro de actualizar los Datos?',  
    text: "Señal visible en la parte de registro de Resultados",  
    icon: 'warning',  
    showCancelButton: true,  
    confirmButtonColor: '#1c044a',  
    cancelButtonColor: '#e1433b',  
    confirmButtonText: 'Si, actualizar'  
  }).then((result) => {  
    if (result.isConfirmed) {  
      const miObjeto = {  
        sampling_id: values.id,  
        results: inputValues,  
        date_sampling: values.date_sampling !== null && values.date_sampling !== undefined ? values.date_sampling : undefined,  
        temperature: values.tempera,  
        presentation: values.presentation,  
        stage: values.stage,  
        status: values.status,  
        date_reception: values.date_reception !== null && values.date_reception !== undefined ? values.date_reception : undefined,  
        date_analysis_start: values.date_analysis_start !== null && values.date_analysis_start !== undefined ? values.date_analysis_start : undefined  
      }  
    }  
  })  
}
```

Figura 38. Código para actualizar las Muestras.

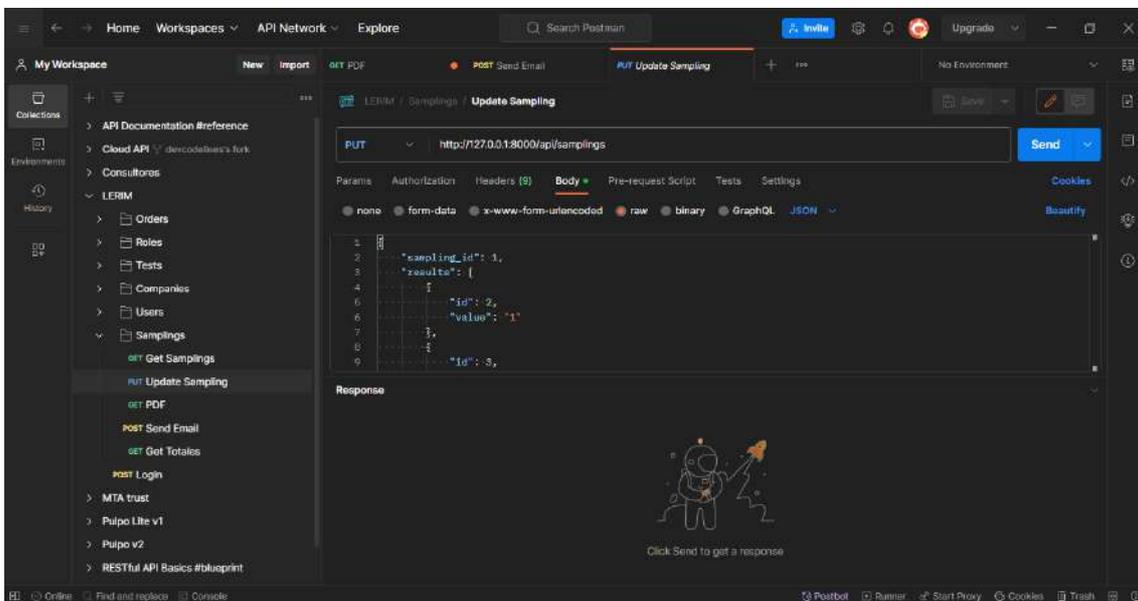


Figura 39. Uso de API para actualizar Muestras.

The screenshot shows a web form titled "Tienda - Se revisara agua de la llave - 2091 - HS/22". The form contains several sections:

- Fechas:** A table with columns for "F. y H. Toma de muestra", "F. y H. Recepción", "F. y H. Inicio análisis", "F. y H. Fin análisis", "Etapas", and "Estatus". The "Estatus" column has radio buttons for "En proceso" and "Entregado".
- Datos:** Input fields for "Temperatura" (35 °C) and "Presentación" (Embotellada).
- Determinaciones:** Input fields for "Salmonella" (Positivo) and "Observaciones" (Dio positivo a Salmonella).

A "Guardar" button is located at the bottom of the form.

Figura 40. Interfaz para actualizar Muestras.

- Ver Muestras

```

const columns = [
  {
    title: 'Folio',
    dataIndex: 'folio',
    key: 'folio',
  },
  {
    title: 'CURI',
    dataIndex: 'curi',
    key: 'curi',
  },
  {
    title: 'Nombre del Cliente',
    dataIndex: 'origin',
    key: 'origin',
  },
  {
    title: 'Descripción',
    dataIndex: 'description',
    key: 'description',
  },
  {
    title: 'Tipo',
    dataIndex: 'type',
  }
];
  
```

Figura 41. Código para ver las Muestras.

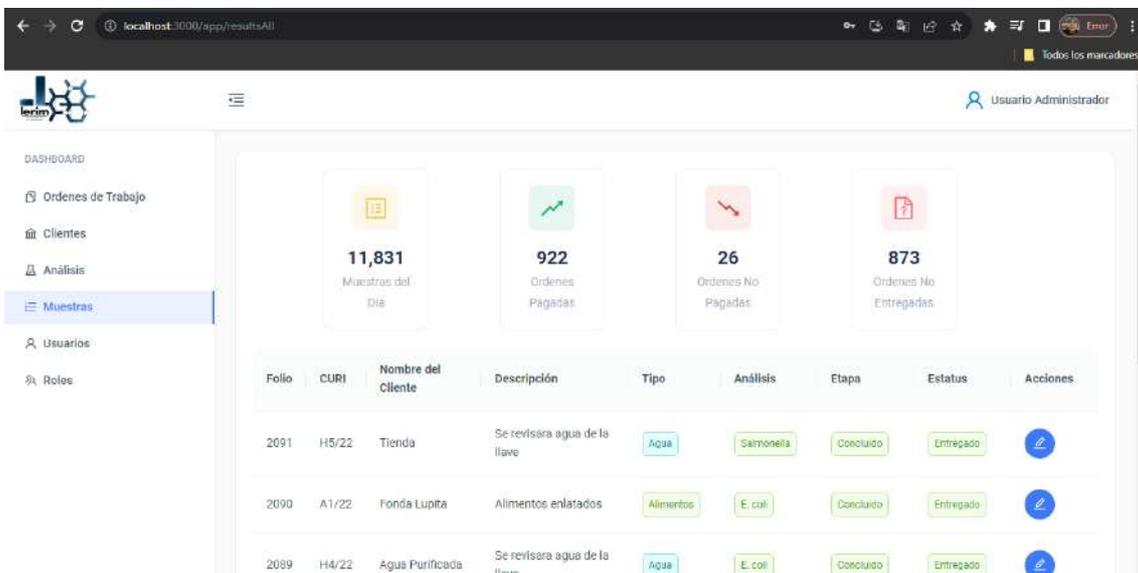


Figura 42. Interfaz para ver las Muestras.

Sprint 5

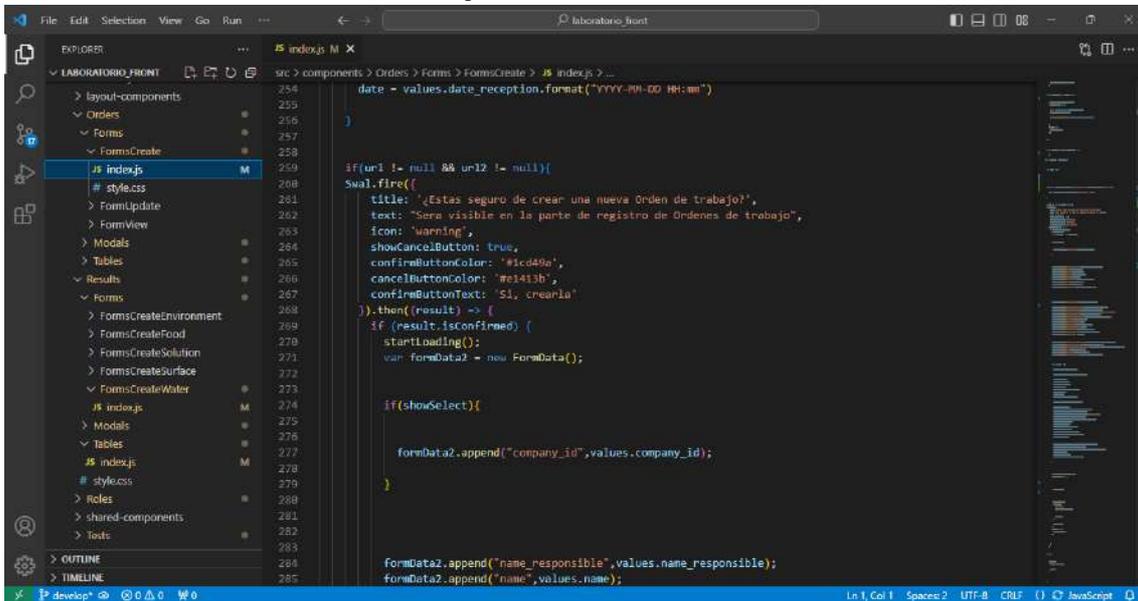
Diseñar la interfaz para el catálogo de ordenes de trabajo con sus respectivas funcionalidades como agregar, modificar y activar o desactivar el status de cada uno de ellos. En la siguiente tabla se muestra el tiempo en que se realizaron las siguientes tareas.

Sprint 5			
No. De Tarea	Duración del sprint	Fecha del Inicio de Sprint	Fecha fin del Sprint
1	3 semanas	13/Noviembre/2023	01/Diciembre/2023
2			
3			
4			
5			

Tabla 7. Sprint 5

Implementación del Sprint 5

- Crear Ordenes de Trabajo



```
date = values.date_reception.format("YYYY-MM-DD HH:mm")
}

if(url1 != null && url2 != null){
  Swal.fire({
    title: '¿Estas seguro de crear una nueva Orden de trabajo?',
    text: 'Sera visible en la parte de registro de Ordenes de trabajo',
    icon: 'warning',
    showCancelButton: true,
    confirmButtonColor: '#1cd496',
    cancelButtonColor: '#e1413b',
    confirmButtonText: 'Si, crearla'
  }).then((result) => {
    if (result.isConfirmed) {
      startLoading();
      var formData2 = new FormData();

      if(showSelect){
        formData2.append("company_id",values.company_id);
      }

      formData2.append("name_responsible", values.name_responsible);
      formData2.append("name",values.name);
    }
  });
}
```

Figura 43. Código para crear las Ordenes de Trabajo.

Orden de Trabajo No. de Folio 2093

¿Nuevo Cliente? Sí No

Nombre del Responsable: Factura: NA

Nombre Comercial: RFC:

Correo Electrónico: Teléfono:

Dirección:

Agregar Fila

Muestras

Fecha y Hora de Muestreo	Tipo	Descripción de la Muestra	Unidades	Análisis	TNC
2023-11-24 19:38:00					

Observaciones

Tipo de Muestreo: Cliente Personal LERIM Remitido al laboratorio

Conservación de Muestra: Congelada Refrigerada Temp. ambiente NA

Figura 44. Interfaz para crear las Ordenes de Trabajo.

- **Ver Ordenes de Trabajo**

```

const columns = [
  {
    title: 'No. folio',
    dataIndex: 'folio',
    key: 'folio',
  },
  {
    title: 'Nombre Comercial',
    dataIndex: ['company', 'name'],
    key: 'companyName',
  },
  {
    title: 'Nombre del Responsable',
    dataIndex: ['company', 'name_responsible'],
    key: 'companyName_responsible',
  },
  {
    title: 'Estatus de Pago',
    dataIndex: 'payment_status',
    key: 'payment_status',
    render: (payment_status) => {
      if(payment_status===1){
        return ( <Tag color="green">Pagado</Tag> )
      }else{
        return ( <Tag color="red">No Pagado</Tag> )
      }
    }
  }
]

```

Figura 45. Código para ver las Ordenes de Trabajo.

Ordenes de Trabajo

No. folio	Nombre Comercial	Nombre del Responsable	Estatus de Pago	Analisis solicitados	Acciones	Estatus
2992	Panaderia	Luis Miguel Hernandez	Pagado	Hongo	[Icons]	[Toggle]
2991	Tienda	Samuel Martinez	Pagado	Semoviente	[Icons]	[Toggle]
2990	Fondo Lupita	Fernando Reyes	Pagado	Ecol	[Icons]	[Toggle]
2989	Agua Purificada	Carlos Menzella	Pagado	Ecol	[Icons]	[Toggle]

Figura 46. Interfaz para ver las Ordenes de Trabajo

Enviar Resultados

* Correo Electrónico

samuel.martinez1122@uppuebla.edu.mx

Enviar

Figura 47. Interfaz para mandar los resultados de las muestras.



Figura 48. Interfaz para confirmar el envío de correo.



Figura 49. Interfaz de mensaje de confirmación del envío del correo.

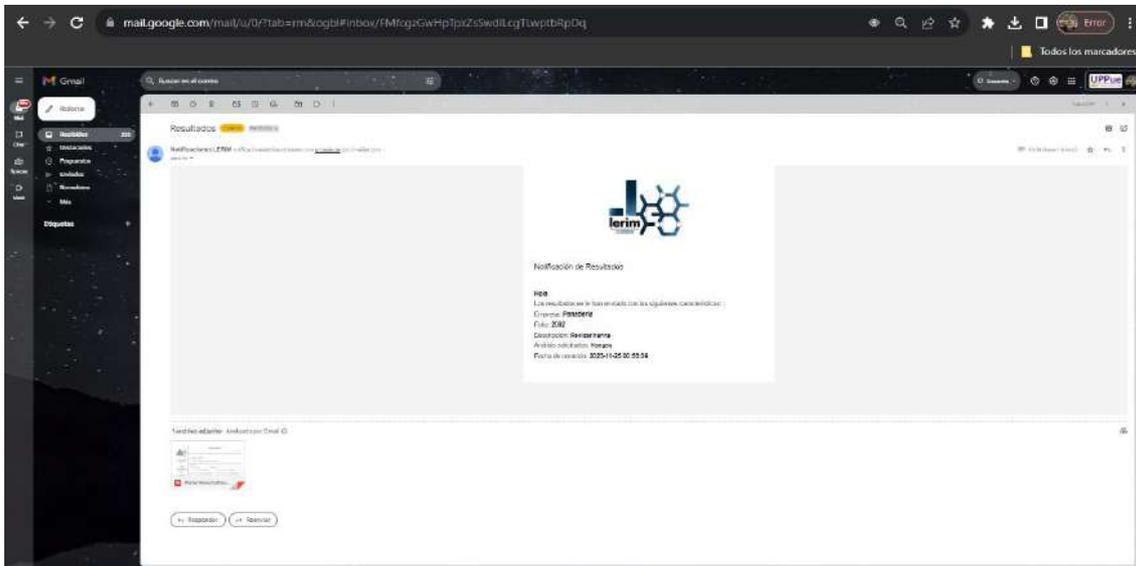


Figura 50. Correo recibido.

3.4 Revisión y Retrospectiva

Para la revisión y retrospectiva se analizó con el Scrum Master y el equipo Scrum.

3.4.1 Revisión de Sprints

Sprint	Elemento Revisado	Logros	Desafíos	Acciones propuestas
1	Instalación de plantilla y diseño del inicio de sesión	Diseño aprobado por el Scrum Master	Fallo en la conexión de la base de datos	Resolución en la base de datos
2	Diseño e implementación de catálogos de usuarios, roles y análisis	Diseño aprobado por el Scrum Master y el equipo Scrum	Correcciones solicitadas por el cliente.	Realizar las correcciones que fueron solicitadas
3	Diseño e implementación del catálogo de clientes	Diseño aprobado por el Scrum Master	Problemas con la funcionalidad	Buscar cual es el problema que hay y solucionarlo
4	Diseño e implementación del catálogo de Muestras	Operaciones CRUD implementadas correctamente	Problemas con la funcionalidad en el procesos de pruebas	Buscar la información necesaria para poder solucionar el error
5	Diseño e implementación del catálogo de	Operaciones CRUD implementadas correctamente y	Correcciones solicitadas por el cliente	Realizar cambios necesarios

	Ordenes de Trabajo	diseño aprobado por el Scrum Master		
--	--------------------	-------------------------------------	--	--

Tabla 8. Revisión de Sprints

3.4.2 Retrospectiva de Sprints

Sprint	Lo bueno	Lo malo	Mejoras
1	Se acepto el diseño de la interfaz	Errores en las consultas con la Base de Datos	Revisar con cuidado cada parte del código
2	Implementación correcta	Respuesta tardía por parte de cliente	Hablar constantemente con el cliente
3	Colaboración correcte por parte del Backend	Errores al realizar las pruebas	Reuniones para revisión de errores
4	Diseño correcto de las interfaces del catálogo	Cambios constantes por parte del Cliente	Darle opciones de diseño al cliente
5	Las pruebas se realizaron correctamente	Respuesta tardía por parte de cliente	Tener una mejor comunicación con el cliente

Tabla 9. Retrospectiva de Sprints

3.5 Lanzamiento

Con el fin de validar los sprints, se ha decidido crear una lista de verificación de criterios de aceptación.

3.5.1 Sprint 1

Tarea	Criterios de aceptación	Completado
Instalación de la plantilla	La instalación se hizo correctamente	X
Diseñar la interfaz para poder iniciar sesión	Se implemento correctamente	X
Hacer conexión a la base de datos para obtener usuario y contraseña	La conexión se hizo de manera exitosa	X
Realizar pruebas	Las pruebas se hicieron correctamente	X

Tabla 10. Verificación de Sprint 1

3.5.2 Sprint 2

Tarea	Criterios de aceptación	Completado
Diseñar catálogos de usuarios, roles y análisis	El diseño fue aceptado	X
Realizar pruebas	Las pruebas se hicieron, y los resultados fueron correctos	X
Hacer las correcciones necesarias	Las correcciones se hicieron y funciono correctamente	X

Tabla 11. Verificación de Sprint 2

3.5.3 Sprint 3

Tarea	Criterios de aceptación	Completado
Diseñar el catálogo de clientes	El diseño se realizó correctamente y fue aceptado	X
Realizar pruebas para revisar la funcionalidad correcta	Las pruebas se realizaron correctamente	X
Hacer correcciones	Las correcciones se hicieron y funciono correctamente	X

Tabla 12. Verificación de Sprint 3

3.5.4 Sprint 4

Tarea	Criterios de aceptación	Completado
Diseño de la interfaz y consultar API de muestras	El diseño fue avalado y aceptado	X
Consultar tabla de análisis para saber que análisis se realizara a la muestra que se entrego	Las consultas se hicieron de manera exitosa	X
Realizar pruebas en la interfaz	Las pruebas se hicieron correctamente	X

Tabla 13. Verificación de Sprint 4

3.5.5 Sprint 5

Tarea	Criterios de aceptación	Completado
Diseño del catálogo ordenes de trabajo.	El diseño se realizó correctamente y fue aceptado	X
Consultar las tablas de muestras y clientes para poder hacer la relación.	Las consultas a la Base de Datos se hicieron de manera correcta	X
Consultar API.	La consulta se hizo correctamente	X
Realizar pruebas de funcionalidad.	Las pruebas se hicieron correctamente	X

Tabla 14. Verificación de Sprint 5

4 Conclusiones y recomendaciones

Quiero resaltar que la adopción de la metodología Scrum en el desarrollo de una aplicación web para un laboratorio microbiológico proporciona agilidad y flexibilidad. Scrum permite adaptarse rápidamente a los cambios en los requisitos del proyecto, lo cual es esencial en entornos donde la investigación y los métodos pueden evolucionar.

Scrum fomenta la colaboración entre los miembros del equipo, incluyendo desarrolladores front-end, lo que es crucial para garantizar que la aplicación web satisfaga las necesidades específicas del laboratorio microbiológico. La comunicación frecuente y transparente facilita la comprensión de los requisitos y expectativas.

Al iniciar la ejecución de un proyecto de desarrollo de software, es imperativo considerar exhaustivamente todos los requisitos y aspectos inherentes al mismo. Esto se traduce en la identificación de una metodología que posibilite la creación del sistema de forma altamente eficiente, sin demoras, y alineada con los estándares y objetivos preestablecidos. La piedra angular de este proceso radica en los requisitos y una planificación precisa, por lo cual subrayo la importancia de dedicar una atención especial y diligente para garantizar la resolución adecuada de estas fases cruciales. Además, la integración de nuevas tecnologías de desarrollo, tal como se emplean en este proyecto específico, puede ofrecer un apoyo sustancial en la creación de sistemas diversos. Por último, enfatizo la recomendación de aplicar esta metodología, especialmente en proyectos propensos a cambios significativos durante el desarrollo. La razón subyacente es su capacidad para proporcionar una programación organizada, reducir la incidencia de errores y otorgar un control de versiones más efectivo, optimizando así su implementación.

5 Referencias bibliográficas

- ebac. (17 de Julio de 2023). *ebac*. Obtenido de ebac: <https://ebac.mx/blog/que-es-react>
- Jesús. (25 de Abril de 2022). *dongee*. Obtenido de dongee: <https://www.dongee.com/tutoriales/que-es-xampp/>
- Juan. (17 de Octubre de 2022). *assemblerinstitute*. Obtenido de assemblerinstitute: <https://assemblerinstitute.com/blog/que-es-postman/>
- Lizarzaburu Pezua, S. (29 de Junio de 2023). *stevenlizarzaburupezua*. Obtenido de stevenlizarzaburupezua: <https://stevenlizarzaburupezua.com/github-que-es-ventajas-y-desventajas/>
- Mancuzo, G. (10 de Julio de 2020). *blog.comparasoftware*. Obtenido de blog.comparasoftware: <https://blog.comparasoftware.com/fases-metodologia-scrum/>
- Mancuzo, G. (17 de Julio de 2020). *blog.comparasoftware*. Obtenido de blog.comparasoftware: <https://blog.comparasoftware.com/scrum-ventajas-y-desventajas/>
- Martins, J. (19 de Junio de 2023). *asana*. Obtenido de asana: <https://asana.com/es/resources/what-is-scrum>
- Perez Hernandez, M. (17 de Octubre de 2017). *mape309site*. Obtenido de mape309site: <https://mape309site.wordpress.com/2017/10/24/ventajas-y-desventajas-de-mysql-phpmyadmin-visual-foxpro-y-microsoft-access/>
- solbyte. (15 de Junio de 2023). *solbyte*. Obtenido de solbyte: <https://www.solbyte.com/blog/nodejs/>



Universidad Politécnica de Puebla
Ingeniería en Informática

Samuel Martínez Arenas
David Ángel Hernández Sánchez
Araceli Ortiz Carranco

Este documento se distribuye para los términos de la
Licencia 2.5 Creative Commons (CC-BC-NC-ND 2.5 MX)