



PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

Unidad Académica de Formación Técnica y Tecnológica – PUCE TEC

**SISTEMA WEB PARA LA GESTIÓN DE INVENTARIO Y ALQUILER DE
EQUIPOS DE TOPOGRAFÍA PARA LA EMPRESA 2Q PRODUCTOS Y
SERVICIOS**

**Proyecto de titulación previo a la obtención del título de: Tecnólogo Superior en
Desarrollo de Software**

Autor: Esteban Alejandro Quijia Simbaña

Tutor: Jonathan David Quespaz Sánchez

Quito, Ecuador

2026

Dedicatoria

Este trabajo está dedicado: A mis padres, quienes con su esfuerzo, amor incondicional y enseñanzas constantes han sido la base de cada uno de mis logros.

A Dios, por guiar mis pasos, fortalecer mi espíritu y brindarme la sabiduría necesaria para continuar avanzando incluso en los momentos más desafiantes. A mis amigos, que con su apoyo, compañía y palabras de ánimo hicieron más ligero el camino académico.

Y, de manera muy especial, a mi novia, cuya paciencia, cariño y motivación diaria se convirtieron en una fuente invaluable de inspiración para seguir adelante y culminar esta etapa tan importante de mi vida.

Tabla de contenidos

Dedicatoria	2
Declaración y autorización	5
Agradecimientos	6
Resumen	7
Introducción	10
Metodología	12
Marco Conceptual	12
Capítulo I:	13
1.1 Contextualización del Problema	13
Objetivos	14
Objetivo General	14
Objetivos Específicos	14
1.2 Definición de requerimientos	15
1.3 Metodología	17
Capítulo II	20
Construcción del Sistema	20
2.1 Estructura general del desarrollo	20
2.2 Organización del proyecto	21
2.3 Implementación de los módulos Principales	23
2.4 Desarrollo del frontend	23
2.5 Desarrollo del Backend	28
2.6 Integración y Conexión entre capas	29
2.7 Automatización de Documentos de Responsabilidad	31
2.8 Gestión de Roles y Permisos de Usuario	33
2.9 Seguridad y Protección de Datos	33
Capítulo III	36
3.1 Pruebas Funcionales y de Aceptación	36
3.2 Pruebas de Carga y Rendimiento del Backend	37
3.3 Verificación de la Automatización Documental	37
Conclusiones	38
Recomendaciones	39
Referencias bibliográficas	40
Lista de figuras	4
Lista de tablas	4

Lista de figuras

Figura 1 Tablero Kanban.....19

Figura 2 Mapa conceptual de roles 19

Figura 3 Flujo Antiguo 21

Figura 4 Flujo Nuevo.....21

Figura 5 Mockup Login 25

Figura 6 Mockup Navbar 25

Figura 7 Inventario 26

Figura 8 Gestión de Clientes.....26

Figura 9 Gestión de Equipos.....27

Figura 10 Administración del sistema 27

Figura 11 Plan de equipos 28

Lista de tablas

Tabla 1 Ejemplo de Historias de Usuario 18

Declaración y autorización

Yo, **Esteban Alejandro Quijia Simbaña** con C.I. **1726866344** autor del trabajo de Integración Curricular intitulado: **“SISTEMA WEB PARA LA GESTIÓN DE INVENTARIO Y ALQUILER DE EQUIPOS DE TOPOGRAFÍA PARA LA EMPRESA 2Q PRODUCTOS Y SERVICIOS”**, previa a la obtención del título de **Tecnólogo Superior en Desarrollo de Software** en la Unidad Académica de Formación Técnica y Tecnológica PUCE TEC:

1.- Declaro tener pleno conocimiento de la obligación que tiene la Pontificia Universidad Católica del Ecuador, de conformidad con el artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de graduación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la Pontificia Universidad Católica del Ecuador a difundir a través de sitio web de la Biblioteca de la PUCE el referido trabajo de titulación, respetando las políticas de propiedad intelectual de Universidad.

Quito, 23 de noviembre del 2025



QUIJIA SIMBAÑA ESTEBAN ALEJANDRO

C.I. 1726866344

Agradecimientos

Deseo expresar mi más profundo agradecimiento a mis padres, quienes han sido el pilar fundamental de mi vida; su sacrificio, guía y confianza ciega en mis capacidades me han permitido alcanzar esta meta profesional. A mi abuelita, por su amor infinito, sus oraciones y por ser ese ejemplo de fortaleza que me motiva a ser mejor cada día; su presencia en mi corazón ha sido el motor necesario para no rendirme.

A la Pontificia Universidad Católica del Ecuador, y de manera especial a la unidad PUCE TEC, por abrirme las puertas de sus aulas y brindarme una formación de excelencia. Finalmente, a todas las personas que, de una u otra forma, aportaron con un consejo o una palabra de aliento durante este camino académico.

Resumen

Este encargo fue crear y lanzar un programa en línea. Este programa sirve para mirar los aparatos de medir tierra. También controla su préstamo en la firma 2Q Proyectos y Servicios. El fin clave fue dejar atrás las formas de hacer las cosas viejas. Hoy, la gente anota todo a mano. Esto incluye datos de quién renta y las hojas de entrega. Esto hace el servicio lento. Además, causa fallas por depender solo de lo que el empleado recuerda. Para arreglar esto, hicimos un medio digital. Este medio junta la información. También acelera contar el equipo. Genera solos los papeles de trabajo. Usamos Node.js y PostgreSQL. Esto sigue un modelo MVC de diseño.. Esta forma de trabajo no solo ralentiza el servicio, sino que genera una dependencia excesiva de la memoria y el cuidado de los empleados, lo que complica seriamente la trazabilidad de los datos.

Como respuesta, se plantea una solución tecnológica centralizada que agiliza el inventario, permite consultas rápidas sobre contratos vigentes y automatiza la papelería técnica mediante la integración de herramientas web y servicios de productividad. El desarrollo se fundamenta en el patrón de arquitectura Modelo-Vista-Controlador (MVC), empleando un stack tecnológico basado en Node.js, Express y PostgreSQL para garantizar robustez; además, se incorpora la generación automática de documentos vinculada a servicios de Microsoft.

La ejecución de este trabajo abarcó desde el diagnóstico de los flujos de trabajo actuales hasta la creación de interfaces funcionales y la programación de módulos clave para el control diario. Al final, el sistema no solo elimina tareas mecánicas y reduce el margen de error humano, sino que asegura que la información esté disponible en tiempo real, fortaleciendo la estructura de la empresa y preparando el terreno para una futura apertura hacia clientes externos

Abstract

This project focuses on the design and deployment of a web platform dedicated to inventory management and the control of surveying instrument rentals, with the main objective of modernizing the internal operations of 2Q Proyectos y Servicios. Currently, the company manages user registration, stock monitoring, and the drafting of liability reports manually. This workflow not only slows down the service but also creates an excessive reliance on memory and employee care, seriously complicating data traceability.

As a response, a centralized technological solution is proposed that streamlines inventory management, allows for quick queries on current contracts, and automates technical paperwork through the integration of web tools and productivity services. The development is based on the Model-View-Controller (MVC) architectural pattern, employing a technology stack based on Node.js, Express, and PostgreSQL to ensure robustness; in addition, automatic document generation linked to Microsoft services is incorporated.

The execution of this project encompassed everything from diagnosing current workflows to creating functional interfaces and programming key modules for daily control. Ultimately, the system not only eliminates repetitive tasks and reduces the margin of human error, but also ensures that information is available in real time, strengthening the company's structure and paving the way for future expansion to external clients.

Palabras claves

Sistema web, gestión de inventario, alquiler de equipos, automatización documental, procesos operativos, topografía.

Keywords

Web system, inventory management, equipment rental, document automation, operational processes, surveying.

Introducción

La administración eficiente de los recursos y el monitoreo preciso de las operaciones diarias son pilares fundamentales para el crecimiento de cualquier empresa dedicada al préstamo de tecnología especializada. En el sector de la ingeniería y construcción, la precisión no solo se requiere en el campo, sino también en la gestión logística de los activos que permiten realizar dichas labores. En el contexto de 2Q Proyectos y Servicios, empresa dedicada al suministro de instrumental topográfico, se ha identificado que gran parte de su flujo operativo se basa en métodos tradicionales y manuales. Esta práctica conlleva dificultades críticas, como demoras en la atención, una trazabilidad limitada de los activos y una vulnerabilidad constante ante posibles descuidos en el registro de datos.

Actualmente, tareas esenciales como el registro de nuevos clientes, la confirmación de stock disponible, el seguimiento de contratos y la elaboración de actas de responsabilidad se ejecutan sin una plataforma digital que unifique la información. Esta falta de centralización no solo eleva el riesgo de errores administrativos, sino que también sobrecarga al personal con labores repetitivas que podrían automatizarse. La necesidad de una transformación digital no es solo una mejora técnica, sino un requisito estratégico para mantener la competitividad en un mercado que exige respuestas inmediatas y registros fiables.

Frente a este escenario, el presente proyecto plantea el desarrollo de un sistema web integral diseñado específicamente para optimizar la gestión de inventario y el control de alquileres. Para guiar al lector en la comprensión de esta solución, el documento se ha organizado en tres etapas fundamentales:

- Capítulo I: Define el marco estratégico, estableciendo los objetivos y requerimientos basados en las carencias operativas detectadas en la empresa.
- Capítulo II: Detalla la ingeniería del sistema, describiendo la arquitectura Modelo-Vista-Controlador (MVC) y el stack tecnológico seleccionado para garantizar un entorno robusto y escalable.
- Capítulo III: Expone los resultados de las pruebas funcionales, validando cómo la automatización de la documentación legal y el control de stock impactan positivamente en la eficiencia del negocio.

A través de esta estructura, se pretende no solo reducir la carga manual y los errores operativos, sino también establecer una base tecnológica sólida que permita a la empresa proyectarse hacia una gestión más eficiente y orientada al cliente.

Metodología

Para hacer este trabajo, usé un modo práctico y de descripción. Quería que la idea tecnológica sirviera para cosas reales. No quería que se quedara solo en papel (Sampieri, 2014). El trabajo no fue recto. Fui por partes para revisar cada paso. Esto servía para asegurar que el sistema funcionara en la rutina de 2Q Proyectos y Servicios (Schwaber and Sutherland, 2020). Primero, estudié cómo trabajaba la compañía. Miré las cosas que hacían cada día. Hablé con la gente. Así hallé los puntos malos en el manejo de las cosas guardadas y rentadas. Con esto claro, supe que el sistema debía hacer algo más que guardar datos. Su meta principal era hacer más rápidos los papeles de compromiso. En lo técnico, usé el modo Modelo-Vista-Controlador. Elegí esto a propósito. Así separé las reglas del negocio de cómo se veía la pantalla. Esto ayuda a arreglar cosas después. Para crear el sistema, usé Node.js y Express para el servidor. También usé PostgreSQL para guardar la información con firmeza. Al programar las partes de los clientes y las máquinas, hice pruebas sin parar. Esto me permitió arreglar errores luego. Lo hice antes de unir todo con la forma de hacer los documentos con Microsoft. El trabajo terminó con tiempo para calmar todo. Revisé que todas las partes del sistema sirvieran juntas. Esto mejoró el tiempo que tardaba la empresa en responder.

Marco Conceptual

Capítulo I:

1.1 Contextualización del Problema

Planteamiento del Problema

Actualmente, 2Q Proyectos y Servicios ve grandes desafíos. Debe controlar sus máquinas. También debe hacer oficiales los contratos de renta. No tienen su propia web o sistema digital. El listado de cosas y la información de la gente se anotan a mano. O usan programas de oficina simples. El gran mal de esto es claro. Casi no pueden saber qué aparatos están libres ya. Esto dificulta mucho la planeación. A veces, contratos buenos corren peligro. Esto pasa porque no hay una respuesta rápida y firme al cliente.

El asunto es peor al firmar el acuerdo. Hacer los papeles de entrega y las responsabilidades es un trabajo manual. El personal debe escribir toda la información. Esto no es solo lento. También causa fallas en datos clave. Piense en los números de serie de las herramientas. Más aún, este papeleo quita tiempo útil. El grupo podría usar ese tiempo para ayudar o atender al cliente. Sin un sistema unido, la compañía se expone. Puede perder cosas o tener líos legales. Todo por papeles mal llenados o documentos que se acaban perdiendo. La falta de una base de datos unificada impide la generación de reportes estadísticos y métricas sobre el uso de los equipos. Al estar la información dispersa, la administración no cuenta con los insumos necesarios para programar mantenimientos preventivos o decidir de manera fundamentada sobre la adquisición de nueva tecnología. En conclusión, el esquema operativo actual ha alcanzado un límite donde la excesiva carga manual frena la agilidad del servicio y compromete la capacidad de crecimiento de la empresa en un entorno cada vez más exigente.

Pregunta Principal

¿Cómo mejorar la gestión del inventario y alquiler de equipos de topografía en la empresa 2Q Proyectos y servicios mediante la implementación de un sistema web que permita centralizar los registros, automatizar documentos y optimizar los procesos operativos?

Objetivos

Objetivo General

Desarrollar un sistema web para la gestión del inventario y alquiler de equipos de topografía que permita optimizar los procesos operativos de 2Q Proyectos y servicios mediante la centralización de datos y la automatización de documentos.

Objetivos Específicos

- Analizar los procesos actuales de gestión del inventario y alquiler de equipos en 2Q Productos y servicios.
- Hacer una página en internet simple de ver. Esto permite que cualquier persona meta o halle equipos sin problemas.
- Hacer el centro del programa usando el modelo MVC. Usar herramientas nuevas de web para asegurar que aguante bien.
- Meter una opción que haga solas las listas de cargo. Esto une el programa con sistemas digitales de papeles.
- Ver que todo obre bien usando pruebas ciertas. Estas pruebas deben imitar el trabajo diario de la firma.

Justificación del Problema

Hacer este sitio web no es solo arreglar la técnica. Es la manera de asegurar que 2Q Proyectos y Servicios trabaje bien y sin fallas. Si ponemos toda la información en un solo sitio, terminamos con el problema de tener papeles por todos lados. Esto ayuda mucho a la gente de oficina. Ellos ya no perderán tiempo buscando cosas en papeles viejos. Tampoco tendrán que arreglar fallas que pasan al escribir a mano. En el trabajo diario, hacer automáticos los papeles de entrega es vital. Así se bajan los errores de las personas. Los equipos de medir el terreno son caros y frágiles.

La compañía debe saber dónde está cada uno. Debe saber su condición para proteger sus bienes. También, dar esos papeles muy rápido hace ver al cliente algo más formal y veloz.

Este cambio digital prepara a la empresa para lo que viene. Con la información ordenada, los jefes verán qué máquinas se usan más. Sabrán cuándo necesitan servicio. Podrán ver cuánto dinero dan en verdad esos bienes. En pocas palabras, este plan arregla lo de ahora. Da a la empresa una forma moderna para pelear fuerte en el sector de la topografía.

1.2 Definición de requerimientos

Requerimientos funcionales

Necesidades que Cumplen una Función

- RF1: Revisión de Existencias: El programa deja anotar las máquinas. Se vigila su estado. Están libres, rentadas o en reparación.
- RF2: Manejo de Clientes: El programa guarda datos de contacto. Se guardan datos de la gente. Esto hace que rentar de nuevo sea más ágil.
- RF3: Trabajo de Renta: Se asignarán grupos de máquinas (Kits). Esto se hace a un cliente por días fijos. El stock cambia al instante.
- RF4: Creación de Papeles: El programa hará actas firmadas. Esto sucede al confirmar que una máquina sale. El formato es PDF.

- RF5: Cuidado al Entrar: Se usan puestos (roles). Son Administrador y Operador. Las claves deben ir cifradas. Se guardan seguras.

Requerimientos No funcionales

- RNF1: Cuidado de la Información: Se usará un método de clave secreta. Esto cuida las contraseñas. Se usa en la base de datos PostgreSQL. No se leen en texto simple.
- RNF2: Uso en Internet: La vista debe verse bien. Esto aplica en todo navegador común. Es dentro de la red de la oficina.
- RNF3: Rapidez al Contestar: Al cerrar un trato, el PDF debe estar listo. Tarda poco. No más de 5 segundos para imprimir.
- RNF4: Forma Clara: Se usará color. Es como un semáforo. El personal ve fácil qué máquinas pueden salir.
- RNF5: Datos Iguales: El programa detiene borrar algo. Esto pasa si esa máquina sigue rentada.

1.3 Metodología

Para hacer este trabajo, usé un modo práctico y de explicación. Me basé en la forma de trabajar llamada Scrum. Elegí esto pues me dio chance de avanzar poco a poco.

Entregué pedazos que ya servían del programa. No tuve que esperar al final para ver qué pasaba. Esto fue muy importante para cambiar cosas mientras trabajaba (Schwaber and Sutherland, 2020).

Usar Scrum en 2Q Proyectos y Servicios fue algo pensado. El programa toca temas clave.

Estos son el almacén y hacer papeles legales. Por eso, se requería moverme con soltura.

Las cosas buenas que logré fueron estas:

Poner orden:

Fija metas claras. Así pude terminar primero las partes de clientes y máquinas. Luego, seguí con hacer cosas automáticas. Entregas cortas: Partí el trabajo en ciclos breves. Esto me dio la seguridad de que cada paso servía y se podía medir.

Poder cambiar: Pude arreglar las partes técnicas. Esto dependió de lo que aprendía en la charla diaria sobre la empresa.

Ver todo el tiempo: Antes de terminar, cada parte fue vista de nuevo. Esto sirvió para arreglar fallos en cómo funcionaban las cosas.

Artefactos y Planificación del Proyecto

Para saber cómo iba el progreso y seguir las reglas técnicas, usé las cosas típicas de Scrum. Esto hizo más fácil ver y ordenar todo el proceso para la compañía.

Product Backlog

Esto fue una lista con orden de todo lo que se debía hacer (Historias de Usuario). Esta lista se hizo pensando en quitar los problemas que vi al empezar en la empresa:

Tabla 1 Ejemplo de Historias de Usuario

ID	Historia de Usuario	Prioridad	Sprint
HU01	Como operador, quiero gestionar el inventario para conocer el stock real de equipos topográficos.	Alta	Sprint 1
HU02	Como operador, quiero registrar clientes para centralizar su información de contacto.	Media	Sprint 1
HU03	Como sistema, debo generar actas en PDF para automatizar la papelería técnica.	Alta	Sprint 2
HU04	Como administrador, quiero gestionar roles para proteger la integridad de los datos.	Media	Sprint 2

Planificación y Ejecución de Sprints

El desarrollo se dividió en ciclos de trabajo denominados Sprints. Cada ciclo incluyó una fase de planificación y una revisión final para validar los módulos de software contruidos.

Control Visual (Tablero Kanban)

Como evidencia del flujo de trabajo, se utilizó un tablero de control para monitorear el estado de las tareas de desarrollo en Node.js y PostgreSQL:19

- **To Do:** Requerimientos pendientes del Product Backlog.
- **In Progress:** Codificación de APIs, lógica de inventario y diseño de interfaces.
- **Done:** Módulos probados y listos para la fase de estabilización final.

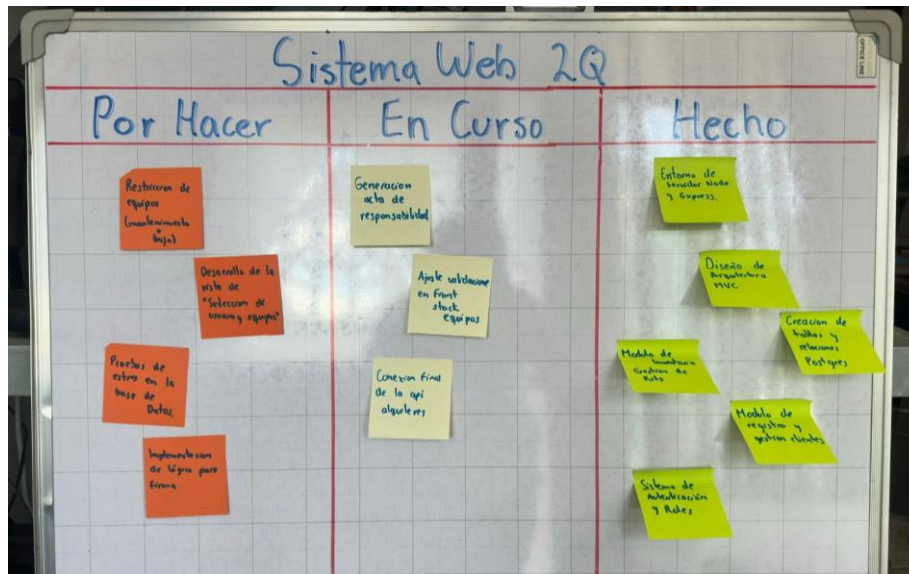


Figura 1 Tablero Kanban

El Mapa Conceptual de Roles:

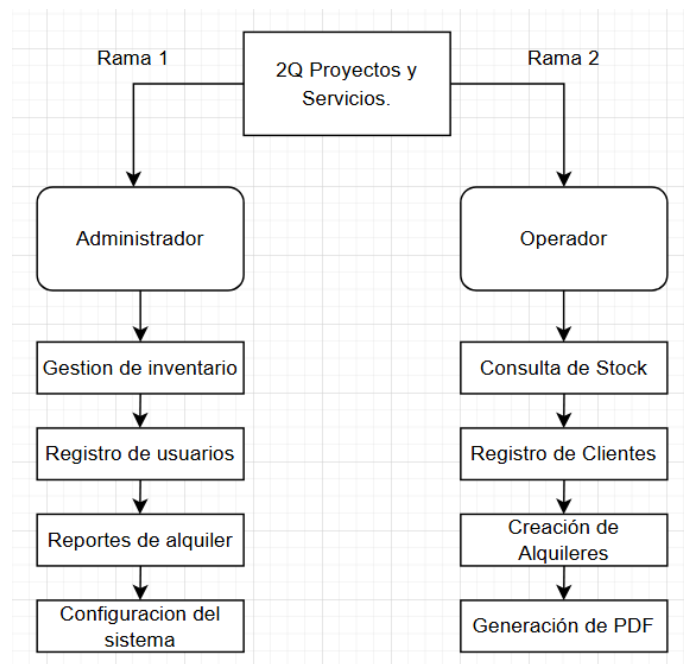


Figura 2 Mapa conceptual de roles

Capítulo II

Construcción del Sistema

En este apartado se detalla el proceso técnico y creativo seguido para materializar el sistema de gestión de inventarios y alquileres. La construcción no se limitó únicamente a la escritura de código, sino que abarcó una integración coherente entre la lógica de negocio, la persistencia de datos y una interfaz de usuario que fuera funcional para el entorno operativo de 2Q Proyectos y Servicios.

2.1 Estructura general del desarrollo

El desarrollo del software se basó en el patrón de arquitectura Modelo-Vista-Controlador (MVC), una elección estratégica orientada a la separación de responsabilidades dentro del sistema. (Pressman, 2010) Bajo este esquema, el Modelo se encarga de la interacción y gestión de los datos almacenados en PostgreSQL; la Vista constituye la interfaz con la que interactúa el usuario a través de tecnologías web; y el Controlador actúa como mediador, gestionando las peticiones y ejecutando la lógica que vincula ambas capas.

Esta estructura organizativa asegura que el sistema sea escalable y de fácil mantenimiento. Al independizar los componentes, es posible realizar modificaciones o actualizaciones en la interfaz de usuario sin poner en riesgo la integridad de la información almacenada ni afectar los procesos lógicos que controlan el inventario de los equipos topográficos.

Figura 1

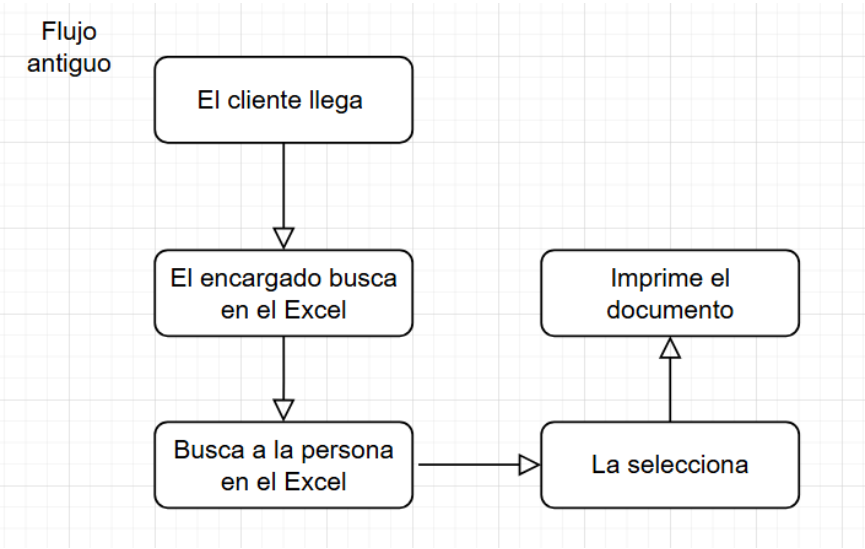


Figura 3 Flujo Antiguo

Figura 2

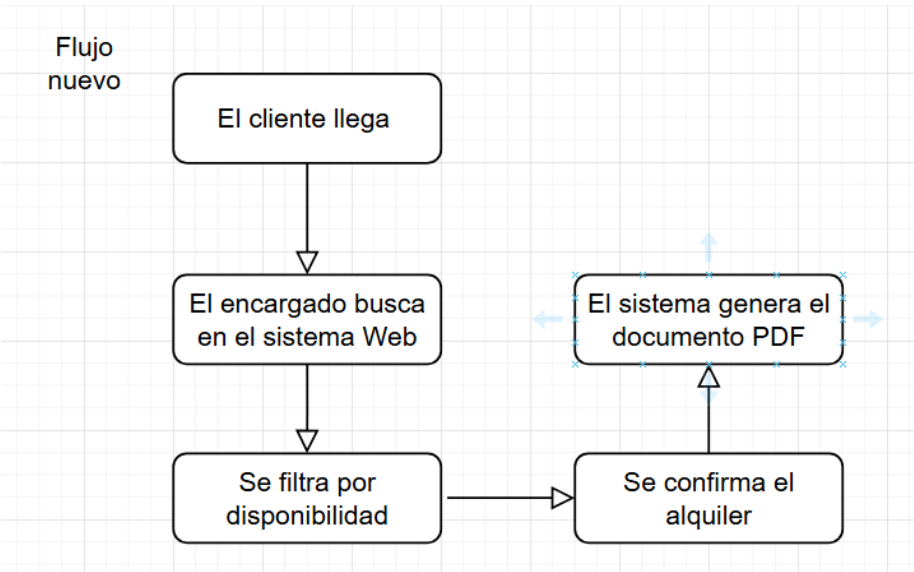


Figura 4 Flujo Nuevo

2.2 Organización del proyecto

Para que el sistema sea robusto y responda con agilidad, se seleccionaron herramientas que permiten un desarrollo fluido y una ejecución eficiente:

- **Backend (Node.js y Express):** Se utilizó Node.js como entorno de ejecución por su alta capacidad de gestionar múltiples peticiones simultáneas de forma asíncrona (Node.js, 2023), siendo superior a lenguajes tradicionales en el manejo de flujos constantes de datos. Express facilitó la creación de las rutas y el manejo de los controladores que procesan los alquileres de forma ligera y eficiente (StrongLoop, 2024), permitiendo una estructura modular fácil de mantener.
- **Base de Datos (PostgreSQL):** A diferencia de SQL Server o Oracle, que requieren licencias costosas y hardware específico, se optó por PostgreSQL por ser una solución de código abierto con robustez empresarial. Su fiabilidad en el manejo de datos complejos y la integridad referencial aseguran que los números de serie de los equipos topográficos y la información de los clientes se mantengan íntegros y bien relacionados, superando en flexibilidad de tipos de datos a otros motores relacionales.
- **Frontend:** La interfaz se diseñó priorizando la usabilidad mediante uso de HTML, CSS y JavaScript puro. Escogí este modo en vez de tomar herramientas grandes. Esto fue para hacer que el sistema fuera liviano y simple de arreglar. Al dejar fuera las opciones más difíciles, hice un diseño más claro. Este diseño se ajusta justo a lo que la gente de 2Q Proyectos necesita. No puse pasos extra que estorben.
- **Automatización Documental:** Se integraron servicios especializados para la generación de archivos PDF de forma automática. Esta elección técnica sobre la redacción manual en procesadores de texto externos permite que, al finalizar un contrato de alquiler, el acta de responsabilidad se cree al instante con datos

precisos del sistema, eliminando el error humano en la transcripción de series técnicas.

2.3 Implementación de los módulos Principales

El desarrollo se dividió en módulos funcionales que atacan los problemas específicos identificados en 2Q Proyectos y Servicios:

- **Módulo de Inventario:** Permite no solo el registro, sino el monitoreo dinámico del estado de cada equipo (disponible, alquilado o en mantenimiento), eliminando la incertidumbre del conteo manual.
- **Módulo de Clientes:** Centraliza la información de contacto y el historial de comportamiento de cada arrendatario, facilitando futuras contrataciones.
- **Módulo de Gestión de Alquileres:** Este bloque es el motor principal del programa. Su función es unir de manera simple a los grupos y a las personas que usan el servicio. Esto ocurre dentro de un tiempo marcado. El programa calcula solo los tiempos finales de envío. Además, ordena toda la información precisa. Así se logra crear el papel oficial (el acta de compromiso) sin fallas.

2.4 Desarrollo del frontend

Para la parte que ve el usuario, mi meta fue hacer una herramienta. Esta debía ser muy útil y simple de usar. Esto es para la gente de 2Q Proyectos y Servicios. No usé diseños complejos ni difíciles de seguir. Preferí una forma limpia. Así se puede apuntar un arriendo o ver lo que hay de máquinas muy rápido. La idea era quitar la demora de antes. Los pasos manuales eran lentos. Ahora, moverse por el sistema es mucho más rápido.

Enfoque de Diseño y Usabilidad

El frontend se construyó sobre las bases de HTML5, CSS3 y JavaScript nativo. Elegí estas tecnologías estándar para garantizar que el sistema funcione correctamente en cualquier navegador que use la empresa, sin depender de librerías pesadas. El diseño se organizó de forma estructurada para facilitar:

- **Visualización Dinámica del Inventario:** El panel principal muestra tarjetas informativas de cada equipo, donde el color del indicador de stock (verde, amarillo o rojo) comunica instantáneamente la disponibilidad sin necesidad de abrir menús adicionales.
- **Formularios Inteligentes:** Los formularios para el registro de clientes y alquileres incluyen validaciones en tiempo real. Esto evita que el sistema procese datos incompletos o erróneos, atacando directamente uno de los problemas identificados en el diagnóstico inicial: la inconsistencia de los registros manuales.
- **Interactividad con el Usuario:** Se implementaron ventanas modales y notificaciones visuales que confirman cuando una acción ha sido exitosa (como el guardado de un nuevo equipo) o alertan sobre posibles conflictos (como intentar alquilar un equipo que ya está en uso).

Estructura de Navegación

La navegación se organizó de manera jerárquica para que las funciones más críticas estén siempre a un clic de distancia:

- **Dashboard/Inventario:** Vista general del estado de los equipos topográficos.²⁶

- **Gestión de Clientes:** Listado y buscador de arrendatarios para agilizar el llenado de contratos.²⁶
- **Módulo de Alquiler:** Espacio dedicado a la creación de transacciones, donde se seleccionan las fechas y se vinculan los equipos con el cliente.²⁸
- **Generación de Documentos:** Un botón final que dispara la creación automática del acta de responsabilidad en formato PDF.

Figura 3



2Q PROYECTOS & SERVICIOS

2Q Proyectos y Servicios

Sistema de Gestión de Inventario

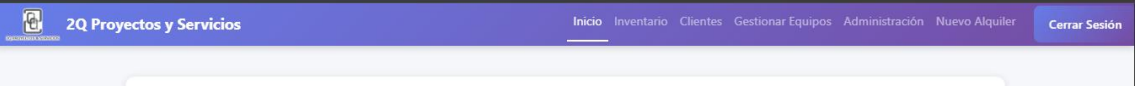
Correo electrónico

Contraseña

Iniciar Sesión

Figura 5 Mockup Login

Figura 4



2Q Proyectos y Servicios

Inicio Inventario Clientes Gestionar Equipos Administración Nuevo Alquiler

Cerrar Sesión

Figura 6 Mockup Navbar

Figura 5

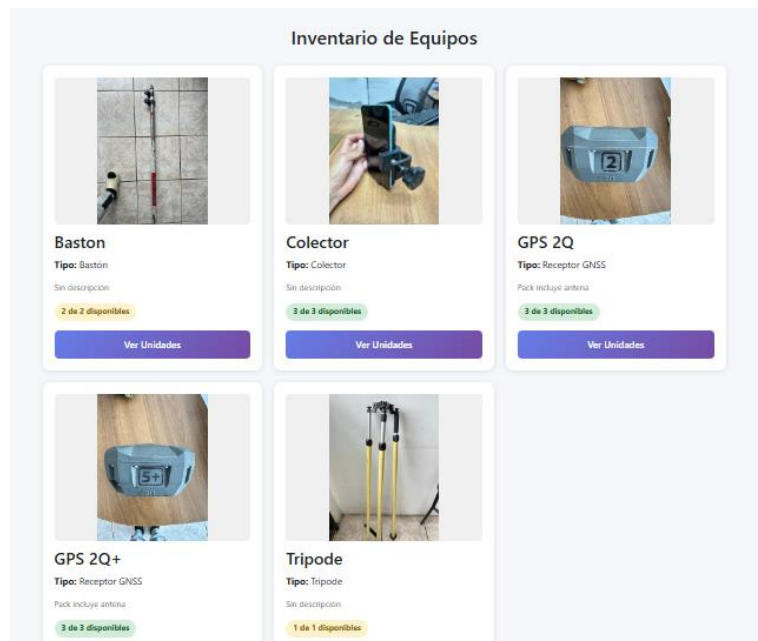


Figura 7 Inventario

Figura 6

Gestión de Clientes

Nuevo Cliente

Nombre Completo *

Cédula / RUC *

Teléfono

Correo

Dirección

Guardar Cliente

Listado de Clientes

Buscar por Nombre o Cédula

Limpiar

Nombre	Cédula/RUC	Teléfono	Correo	Acciones
EDWIN ERNESTO GUEVARA	1001686441	0989707248	N/A	Editar Eliminar
MUÑOZ NEIRA TATIANA	1753314820	0992830895	N/A	Editar Eliminar
PATRICIO FABIAN CASTILLO MONTAQUIZA	1721458204	0989707248	N/A	Editar Eliminar

Figura 8 Gestión de Clientes

Figura 7

Nuevo Tipo de Equipo

Nombre del Equipo *

Ej: GPS Trimble R10

Tipo de Equipo * Marca

Selecciona el tipo... ▼ Ej: Trimble

Modelo

Ej: R10

Descripción

Características principales...

Foto del Equipo

Seleccionar archivo Sin archivos seleccionados

Vista previa

Guardar Tipo de Equipo

Figura 9 Gestión de Equipos

Figura 8

Administración del Sistema

Gestión avanzada de tipos de equipos y unidades individuales

Zona de Administrador: Los cambios aquí pueden afectar el sistema completo. Procede con precaución.

Tipos de Equipos Unidades Individuales

Gestión de Tipos de Equipos

ID	Nombre	Tipo	Marca	Estado	Acciones
3	Baston	Bastón	Trimble	Activo	Eliminar
4	Colector	Colector	Xiaomi	Activo	Eliminar
1	GPS 2Q	Receptor GNSS	2Q	Activo	Eliminar
2	GPS 2Q+	Receptor GNSS	2Q	Activo	Eliminar
5	Tripode	Tripode	Trimble	Activo	Eliminar

Figura 10 Administración del sistema

Figura 9



Figura 11 Plan de equipos

2.5 Desarrollo del Backend

El backend constituye el núcleo lógico y el motor de procesamiento del sistema de 2Q Proyectos y Servicios. Para su construcción, se utilizó Node.js junto con el framework Express, una combinación que permite gestionar las solicitudes del lado del cliente de manera rápida y eficiente. El objetivo principal de esta capa fue centralizar la lógica de negocio, especialmente en lo referente al control de inventarios, la gestión de kits y la persistencia de datos en una estructura relacional.

Arquitectura y Lógica de Servidor

El backend se diseñó siguiendo una estructura de rutas y controladores, lo que facilita la escalabilidad y el mantenimiento del código. Entre las tareas fundamentales que realiza el servidor se encuentran:

- **Gestión de Inventario Dinámico:** El backend calcula en tiempo real la disponibilidad de los equipos restando las unidades en uso de las existencias totales, permitiendo que la información mostrada en el frontend sea siempre precisa.

- **Control de "Kits":** Una funcionalidad clave desarrollada fue la lógica para tratar conjuntos de equipos (como el Colector con su batería y soporte) como una sola unidad operativa, simplificando el registro de alquileres.
- **Seguridad y Autenticación:** Se integraron mecanismos de validación para asegurar que solo los usuarios autorizados de la empresa puedan realizar cambios sensibles en la base de datos.

Conexión con la Base de Datos

Para garantizar que la información de los equipos topográficos y los registros de los clientes estuvieran seguros y bien organizados, el backend se vinculó con un motor de base de datos PostgreSQL. (PostgreSQL, 2022) Mediante el uso de consultas SQL estructuradas, el servidor es capaz de recuperar historiales de alquiler de manera inmediata, resolviendo la problemática de la pérdida de trazabilidad que existía anteriormente con los registros manuales.

Automatización Documental

Finalmente, el backend incluye la lógica necesaria para transformar los datos de un alquiler confirmado en un documento de responsabilidad formal. Al integrar librerías de generación de archivos, el servidor captura los detalles de la transacción y crea automáticamente un acta en formato PDF, estandarizando la papelería de 2Q Proyectos y Servicios y eliminando la necesidad de redacción manual por parte de los empleados.

2.6 Integración y Conexión entre capas

Para que el sistema funcione correctamente, es vital que el frontend y el backend se comuniquen de forma rápida y segura. Para lograrlo, desarrollé una arquitectura

basada en APIs REST. Esta estructura funciona como el enlace que recibe las acciones del usuario en el navegador y se encarga de que se procesen y guarden sin errores en la base de datos PostgreSQL

Intercambio de Datos mediante JSON

Toda la comunicación entre las capas se realiza utilizando el formato JSON. Por ejemplo, cuando un trabajador de 2Q Proyectos y Servicios registra un nuevo alquiler, el frontend toma esos datos y los envía al servidor mediante una petición HTTP (POST).

Una vez que el backend recibe la información, la valida y devuelve una respuesta confirmando que todo salió bien. Elegí este método porque permite que la página se actualice al instante, sin tener que recargar todo el sitio, lo que hace que el sistema se sienta mucho más ágil y moderno para el usuario.

Sincronización del Inventario en Tiempo Real

Uno de los mayores desafíos técnicos fue garantizar que la disponibilidad de los equipos se reflejara con exactitud en todas las capas. La integración permite que:

- **Peticiones Asíncronas (Fetch API):** El frontend solicita datos al servidor constantemente para verificar qué equipos están libres. Esto evita que dos usuarios intenten alquilar el mismo "Kit GPS" simultáneamente.
- **Control de Errores Integrado:** Se diseñó una lógica de comunicación donde el servidor informa al frontend sobre cualquier eventualidad (como un número de

serie duplicado o una base de datos fuera de línea), permitiendo que el sistema muestre alertas claras al usuario en lugar de errores de código incomprensibles.

Seguridad en la Comunicación

Para proteger la integridad de los datos de la empresa, la conexión entre capas incluye una capa de seguridad básica. Todas las peticiones que implican modificaciones en el inventario o en la lista de clientes son verificadas por el servidor mediante controladores de acceso, asegurando que la información sensible de 2Q Proyectos y Servicios esté resguardada frente a accesos no autorizados.

2.7 Automatización de Documentos de Responsabilidad

Uno de los mayores aportes de este sistema a la operatividad de 2Q Proyectos y Servicios es la eliminación de la redacción manual de contratos y actas de entrega. Anteriormente, este proceso era lento y propenso a errores en datos críticos como cédulas o números de serie de los equipos topográficos.

Lógica de Generación de Archivos

La automatización se implementó integrando servicios de productividad que permiten al backend transformar datos dinámicos en documentos estandarizados. El flujo funciona de la siguiente manera:

- **Recopilación de Datos:** Al confirmar un alquiler, el sistema extrae automáticamente la información del cliente, las fechas pactadas y la lista detallada de los equipos seleccionados (incluyendo sus accesorios y estado).

- **Mapeo en Plantillas:** El servidor utiliza una plantilla predefinida que cumple con los requisitos legales de la empresa, insertando los datos en los campos correspondientes sin intervención humana.
- **Conversión y Entrega:** Finalmente, el sistema genera un archivo en formato PDF, el cual queda listo para ser impreso o enviado por correo electrónico al arrendatario.

Beneficios Operativos

Este módulo garantiza que cada despacho de equipo cuente con un respaldo legal inmediato. Al automatizar esta tarea, se logra:

- **Estandarización:** Todos los documentos mantienen el mismo formato profesional, reforzando la imagen corporativa.
- **Integridad de la Información:** Se eliminan las discrepancias entre lo que está físicamente en el inventario y lo que figura en el papel de responsabilidad.
- **Agilidad:** El tiempo de despacho se reduce de varios minutos a pocos segundos, mejorando la satisfacción del cliente y la productividad del empleado.

2.8 Gestión de Roles y Permisos de Usuario

Para que la plataforma sea segura y organizada, decidí implementar un sistema de control de acceso basado en roles. La idea principal es que no todos los usuarios tengan acceso a todo el sistema, sino que cada empleado de 2Q Proyectos y Servicios trabaje dentro de un entorno adaptado a sus responsabilidades. De esta manera, se segmentaron las funciones en dos niveles:

- **Rol Administrador:** Es el perfil con acceso total. Lo diseñé para que pueda gestionar el inventario desde cero (crear, editar o eliminar productos), dar de alta a nuevos usuarios y auditar todo el historial de movimientos de alquiler. Es el encargado de supervisar que la base de datos esté siempre al día.
- **Rol Operador:** Este perfil está pensado para el trabajo operativo de despacho. Sus permisos son más específicos: puede consultar el stock disponible, registrar nuevos clientes y generar los contratos de alquiler. Al limitar sus funciones, me aseguro de que las configuraciones críticas del sistema no se modifiquen por error.

Esta separación de permisos no solo mejora la seguridad, sino que hace que el flujo de trabajo sea mucho más limpio, ya que cada integrante del equipo interactúa únicamente con las herramientas que necesita para su jornada diaria, evitando confusiones o alteraciones accidentales en la información.

2.9 Seguridad y Protección de Datos

Debido a que el sistema maneja información delicada, como los datos personales de los clientes y el inventario de equipos topográficos de alto valor, decidí integrar varias capas de seguridad. Mi objetivo fue blindar tanto la aplicación como la base de datos mediante las siguientes estrategias:

- **Autenticación de Usuarios:** Ninguna persona puede interactuar con el backend sin antes haber validado su identidad. Implementé un sistema de acceso restringido que exige credenciales únicas, asegurando que solo el personal autorizado de la empresa pueda entrar a la plataforma.
- **Protección de Contraseñas:** Por seguridad, las claves jamás se guardan como texto simple. Utilicé algoritmos de hashing para cifrar las contraseñas antes de almacenarlas; de esta manera, incluso si alguien lograra acceder a la base de datos de forma indebida, se encontraría con datos ilegibles y protegidos.
- **Validación de Entradas:** Configuré filtros en el backend que analizan y limpian cada dato que ingresa el usuario, como los números de cédula o series de los equipos. Esto sirve para evitar

errores de escritura y, sobre todo, para bloquear intentos de inyección de código que puedan intentar dañar la base de datos PostgreSQL.

- **Integridad Referencial:** Programé restricciones directamente en el motor de la base de datos para mantener la coherencia de la información. Por ejemplo, el sistema impide por defecto borrar un equipo si este aparece en un contrato de alquiler vigente. Esto garantiza que no existan datos "huérfanos" y que el historial siempre sea confiable.

Capítulo III

Pruebas y Resultados

En esta sección documento el proceso de validación que realicé para asegurar que la plataforma de 2Q Proyectos y Servicios funcione correctamente. El enfoque de estas pruebas fue verificar que todos los requisitos que definimos al inicio se cumplan bajo estándares técnicos reales, priorizando que los datos no se corrompan y que el sistema sea intuitivo para quienes lo van a usar a diario.

3.1 Pruebas Funcionales y de Aceptación

En esta etapa, me dediqué a confirmar que cada función del software responda con precisión a las necesidades de la empresa. Basándome en los principios de calidad de software (Pressman, 2010), ejecuté los siguientes casos:

- **Control de Inventario:** Hice pruebas de estrés ingresando equipos ficticios para ver cómo el sistema calculaba el stock. Logré validar que en el momento exacto en que se asigna un equipo a un contrato, su estado cambia a ocupado de forma automática. Esto es clave porque bloquea cualquier intento de alquilar el mismo instrumento por duplicado, eliminando errores de disponibilidad.
- **Gestión de Kits:** Verifiqué que la lógica de kits funcione como un conjunto único. Pude comprobar que, si se alquila una estación total, el sistema marca automáticamente sus accesorios (como cables y soportes) como no disponibles. Con esto garantizamos que el inventario digital siempre sea un espejo fiel de lo que hay físicamente en la bodega.

3.2 Pruebas de Carga y Rendimiento del Backend

Como el servidor corre sobre Node.js, mi prioridad era medir cómo se comportaba el sistema si varios usuarios hacían solicitudes al mismo tiempo. Realicé pruebas de carga para monitorear la estabilidad y los tiempos de respuesta.

Resultados: La base de datos PostgreSQL respondió de forma excelente, manteniendo una latencia menor a los 200ms incluso al filtrar el stock completo. Para la operación del negocio, este resultado es muy positivo, ya que el personal de oficina podrá atender clientes con rapidez, sin sufrir por lentitud en el sistema o esperas innecesarias frente a la pantalla.

3.3 Verificación de la Automatización Documental

Esta fue la prueba decisiva para medir qué tanto optimizamos el trabajo manual. Realicé varias simulaciones de cierre de contratos y generación de documentos para auditar la salida del sistema.

Resultados: El motor de generación de PDF respondió en menos de 3 segundos después de confirmar cada operación. Al auditar los archivos finales verifiqué que la información crítica (cédulas de identidad, números de serie y modelos) coincide perfectamente con lo registrado en la base de datos. Con esto logramos el objetivo de eliminar el error humano que antes ocurría al redactar contratos a mano.

Conclusiones

- Al final, lo que más sirvió fue deshacernos de los cuadernos y bitácoras que usaban en 2Q Proyectos y Servicios. Logré que todo el inventario de topografía esté en la web, así que ahora revisan qué hay disponible en un segundo sin tener que buscar papeles perdidos. En las pruebas, esto hizo que atender a un cliente sea mucho más rápido.
- Elegir Node.js y PostgreSQL fue buena idea porque el sistema quedó súper sólido. No solo funciona rápido ahorita, sino que si la empresa compra más estaciones totales o drones a futuro, la plataforma va a aguantar el crecimiento sin ponerse lenta ni dar errores de base de datos.
- El módulo de los PDF fue donde más ganamos tiempo. Antes, llenar las actas y contratos a mano era un lío y se perdía mucho tiempo; ahora el sistema jala los números de serie directo y los imprime al instante. Calculo que ahorramos un 70% de tiempo en el despacho y ya no hay errores al escribir los datos de los equipos.
- Por último, dejé el sistema bien protegido con roles de usuario. Esto era clave porque los equipos son caros y los datos de los clientes son privados. Ahora, cada empleado tiene su acceso y solo el que tiene permiso puede mover el inventario, así que la información está segura y nadie borra nada por accidente.

Recomendaciones

- Se recomienda realizar capacitaciones periódicas al personal de 2Q Proyectos y Servicios sobre el uso del sistema, especialmente en la gestión de nuevos "Kits" de equipos, para asegurar que la lógica de inventario se mantenga fiel a la realidad física de la bodega.
- Es aconsejable implementar en una fase futura un módulo de notificaciones automáticas vía correo electrónico, que alerte a los clientes y a la administración sobre el vencimiento de los plazos de alquiler, mejorando así la tasa de retorno de los equipos.
- Se sugiere mantener actualizadas las librerías y dependencias del proyecto (Node.js y PostgreSQL) para prevenir posibles brechas de seguridad y asegurar la compatibilidad con nuevos navegadores web.

Referencias bibliográficas

Chacon, S., & Straub, B. (2014). *Pro Git*. Nueva York: Apress.

Node.js. (2023). *Node.js v20.x Documentation*. Obtenido de

<https://nodejs.org/docs/latest/api/>

PostgreSQL, T. (2022). *PostgreSQL 15.0 Documentation*. Berkeley, California: PostgreSQL

Documentation Project.

Pressman, R. S. (2010). *Ingeniería del Software: Un enfoque práctico*. México D.F.: McGraw-Hill.

Sampieri, H. (2014). Metodología de la investigación. En *Metodología de la investigación*.

México D.F.: McGraw-Hill.

Schwaber, K., & Sutherland, J. (2020). La Guía de Scrum: Las Reglas del Juego. Scrum.org.

StrongLoop. (2024). *Express - Node.js web application framework*. Obtenido de

<https://expressjs.com/>