



**UNIVERSIDAD
DE ANTIOQUIA**

**DESARROLLO DE UNA APLICACIÓN WEB
PARA EL MANEJO INTEGRAL DE LA
INFORMACIÓN DEL LABORATORIO DE
BIOINGENIERÍA**

Autor
Juan Felipe Rayo Puerta

Universidad de Antioquia
Facultad de Ingeniería, Programa de Bioingeniería
Medellín, Colombia
2019





**DESARROLLO DE UNA APLICACIÓN WEB PARA EL MANEJO INTEGRAL DE LA
INFORMACIÓN DEL LABORATORIO DE BIOINGENIERÍA**

Autor:

Juan Felipe Rayo Puerta

Programa de Bioingeniería

Asesor:

Angelower Santana Velasquez

Universidad de Antioquia

Medellín, Colombia

2019

DESARROLLO DE UNA APLICACIÓN WEB PARA EL MANEJO INTEGRAL DE LA INFORMACIÓN DEL LABORATORIO DE BIOINGENIERÍA

Resumen

El Laboratorio de Bioingeniería de la Universidad Pontificia Bolivariana se encontraba en la necesidad de mejorar su sistema de gestión para registrar y visualizar información asociada a los equipos, reactivos, consumibles, implementos y residuos derivados de sus diferentes procesos.

Para suplir esta necesidad, se desarrolló la aplicación web Limbio como plataforma para el manejo de la información del Laboratorio de Bioingeniería desde las 3 diferentes líneas de investigación: Ingeniería de Tejidos, Biomecatrónica y Biomateriales, con las tecnologías de MongoDB, Express, React, Node, y se realizó el montaje de la infraestructura en la intranet de la Universidad Pontificia Bolivariana, permitiendo el acceso a la plataforma bajo el subdominio <http://limbio.upb.edu.co>.

Además, se realizó el Manual de usuario y Guía de instalación de la plataforma, y finalmente, se realizó una validación por usabilidad de la aplicación encontrando resultados favorables en la percepción de 5 usuarios.

Introducción

Un Sistema de Gestión de Información de Laboratorio (*LIMS* por sus siglas en inglés) se define como una herramienta tipo software que administra la información producida o utilizada en laboratorios y en sus diferentes procesos específicos [1]. Seleccionar un *LIMS* adecuado es crítico para las organizaciones y laboratorios que buscan mejorar, complementar o modificar su actual operatividad en el entorno laboral e investigativo. Algunos factores a tener en cuenta para seleccionar el *LIMS* adecuado son: ancho de banda de la red, desempeño, acceso, seguridad, facilidad de uso y costos de hardware y software.

La opción seleccionada, luego de evaluarse las necesidades del Laboratorio de Bioingeniería, es el *Thin-Client* o "de cliente ligero"; esta selección se debe a que el *Thin-Client* trae algunas fortalezas como un acceso ligero y sencillo al *LIMS*, pues no hay necesidad de hardware extra [2]. Además, se realiza el mantenimiento y actualización del *LIMS* desde el servidor, por lo que reduce su dificultad y costo al realizar un soporte técnico en todos los dispositivos que puedan acceder al sistema [2].

Por otra parte, sus limitaciones, como el acceso en tiempo real al servidor para la funcionalidad del usuario, no representan un problema a la hora de implementar el *LIMS*, pues los diferentes equipos de cómputo en el Centro de Bioingeniería y en el Laboratorio de Bioingeniería tienen acceso a la web de forma ilimitada. En este contexto, se utiliza la arquitectura cliente-servidor la cual, empleando navegadores web, accede a la aplicación mediante

direcciones enlazadas a un servidor alojado bien sea en internet o intranet [3].

Con el fin de realizar la plataforma para el manejo de la información del laboratorio, se propone el desarrollo completo de una aplicación web compuesta por la capa de presentación de la misma, conocida como *front-end*, y por las capas de lógica de negocios y de acceso de datos, conocidas como *back-end* [4]. En la arquitectura a emplear, el cliente se considera el *front-end* y el servidor junto a la base de datos se consideran el *back-end* [4]. La comunicación entre el cliente y el servidor se realiza mediante el protocolo HTTP.

Objetivos

Objetivo General

- Diseñar e implementar un Sistema de Gestión de Información web para el laboratorio de Bioingeniería del Centro de Bioingeniería.

Objetivos Específicos

- Desarrollar la capa de presentación, de lógica de negocios y de acceso de datos del sistema en una arquitectura cliente – servidor.
- Implementar un modelo de infraestructura basado en un servicio de computación en la nube y desplegar la aplicación web desarrollada.
- Validar los módulos de la aplicación para garantizar el funcionamiento integral del sistema.

Marco Teórico

Sistemas de Gestión de Información de Laboratorio (LIMS)

Los LIMS son herramientas basadas en software que mejoran el manejo de la información de los procesos de los laboratorios. Para la construcción de estos sistemas se emplean diversas distribuciones y arquitecturas; entre estas la arquitectura "*thin client*" o de cliente ligero, la cual ofrece acceso a todas las funcionalidades de la aplicación mediante el navegador web sin instalaciones necesarias desde el cliente [2]; dicha arquitectura es ampliamente usada debido a su bajo costo de implementación y la facilidad de mantenimiento de la aplicación desde el servidor [5].

Arquitectura cliente-servidor

Es un modelo de diseño de software que genera la división de la aplicación en dos partes: cliente y servidor. Este modelo de red de computación consiste en la solicitud de peticiones por parte de los clientes hacia el servidor, el cual recibe y responde al proceso solicitado [6] empleando el protocolo de transferencia de hipertexto (HTTP) para la comunicación entre las dos partes [3].

JavaScript

Lenguaje de programación interpretado orientado a objetos, utilizado para el desarrollo del cliente (*front-end*) y la implementación del servidor (*back-end*) [7].

Visual Studio Code

Es un editor de código fuente gratuito y de código abierto que permite el desarrollo y depuración de aplicaciones web [8], compatible con varios lenguajes de programación y es reconocido como la herramienta de entorno de desarrollo más popular del 2018 entre los programadores [9].

Front-end

Término que hace referencia a la capa de presentación del software, el cual en la arquitectura presentada es equivalente al cliente [10]. Esta capa permite la vinculación y manipulación de lenguajes de marcado como HTML, de estilo como CSS y de programación como JavaScript. Para la implementación de estos lenguajes existen distintas bibliotecas y *frameworks* que facilitan la construcción de la interfaz de usuario para el cliente como ReactJS, una biblioteca de JavaScript basada en componentes para la construcción de interfaces de usuario [11].

Mockup

Es un modelo que permite la visualización preliminar de un diseño o prototipo de interfaz de usuario de una aplicación web. Para la creación de los mockups pueden emplearse diferentes softwares entre estos el *Adobe XD*, el cual es un software de diseño especializado para la creación de prototipos de interfaces de usuario [12].

Back-end

Término que hace referencia a la capa de lógica de negocios y de acceso de datos permitiendo el alojamiento y ejecución de los programas que conforman la aplicación [10]. En la capa de lógica de negocios se reciben las peticiones del cliente y se envían las respuestas derivadas de un proceso mediante el desarrollo de un servidor empleando un framework como Node.JS, y en la capa de acceso de datos se almacena la información y se permite el manejo de esta [4] haciendo uso de un gestor de bases de datos como el MongoDB.

- Node.JS: entorno en tiempo de ejecución de JavaScript que permite realizar las acciones necesarias para ejecutar un programa escrito en este lenguaje de programación, así como el desarrollo de la capa lógica de negocios de la aplicación web [13].
- MongoDB: sistema de base de datos documental de código abierto, el cual tiene la capacidad de realizar consultas empleando JavaScript [14].

Metodología

Inicialmente, se realizaron esquemas o “*mockups*” de la plataforma para la línea de trabajo de Ingeniería de Tejidos del Laboratorio, para los cuales se empleó la herramienta *Adobe XD* con el fin de generar el prototipo de la interfaz de la plataforma con un diseño centrado en el usuario.

Luego, para desarrollar la aplicación web, se hizo uso del editor de código fuente *Visual Studio Code* mediante el lenguaje de programación orientado a objetos JavaScript, con el cual se utilizó la biblioteca *ReactJS* para la construcción de la interfaz de usuario. Además, se empleó el entorno de ejecución *Node.js* para la creación del servidor web en la capa de acceso de datos utilizando *MongoDB* como sistema de base de datos documental.

Así, al finalizar el desarrollo del sistema para una sola línea de trabajo (Ingeniería de Tejidos) se extendió la aplicación para tener acceso a tres líneas diferentes: Ingeniería de Tejidos, Biomateriales y Biomecatrónica.

Finalmente, se implementó una infraestructura bajo un subdominio de la Intranet de la Universidad Pontificia Bolivariana y se realizó el despliegue de la aplicación en un servidor Windows Server 2016 Desktop Experience.

1. Desarrollo del sistema

Programación de las capas de presentación, de lógica de negocios y de acceso de datos

Para generar la interfaz de usuario se desarrolló el *front-end* empleando la biblioteca *ReactJS* de JavaScript y el entorno de ejecución *Node.js* para el *back-end* de cada módulo. Este proceso se llevó a cabo mediante las siguientes etapas.

1.1. Inicio de sesión y pantalla principal

En esta primera etapa se realizó el *login* de la aplicación, el cual le permite al usuario el acceso a la pantalla principal del sistema dependiendo de la línea de investigación a la que pertenece.

La pantalla principal incluyó dos componentes: una ventana de notificaciones, la cual permitió tener un seguimiento de los eventos asociados a fechas de garantía de equipos, mantenimientos asignados de estos, vencimiento de reactivos, disminución de insumos y consumibles, y de fechas de descarte de los residuos; y, por otra parte, una ventana de anuncios, los cuales son generados por usuarios de tipo administrador.

1.2. Equipos

En este módulo se implementó la visualización del listado de equipos pertenecientes al Laboratorio de Bioingeniería (601E – Ingeniería de

Tejidos) con su respectiva ficha técnica, guía de uso rápido y asignación de mantenimientos.

En este módulo se incluyeron algunas funciones como búsqueda de equipos, agregar nuevos equipos, asignación de mantenimientos, notificación de próximas calibraciones y la generación del listado exportable de los equipos.

1.3. Reactivos

En este módulo se dio acceso a la base de datos de los reactivos del Laboratorio de Bioingeniería (601E – Ingeniería de Tejidos) incluyendo la ficha de seguridad de estos. Adicionalmente, se implementaron los botones para agregar un nuevo reactivo o una nueva alícuota, la visualización de notificaciones con respecto a la disminución de los reactivos en el stock del laboratorio, agotados, vencidos o próximos a vencer. De igual forma, en el módulo de reactivos se le permitió al usuario administrador la generación del listado exportable de los reactivos.

1.4. Consumibles

En esta ventana se suministró el acceso al listado de consumibles (e.g. bolsas de algodón, batas desechables) presentes en el Laboratorio de Bioingeniería (601E – Ingeniería de Tejidos). En este módulo se implementaron funciones tales como: crear nuevo consumible, generación de notificaciones, registro de usos, eliminación de productos y generación del listado exportable de los consumibles.

1.5. Implementos

Para esta fase se generó el listado de implementos (e.g. agitador de vidrio, balones volumétricos) disponibles en el Laboratorio de Bioingeniería (601E – Ingeniería de Tejidos), permitiendo el ingreso de nuevos implementos al sistema, la eliminación de elementos, el reporte de daños y la generación del listado exportable de los implementos.

1.6. Residuos

Para el desarrollo de éste módulo se realizó una base de datos de los residuos derivados de los procesos realizados en el Laboratorio de Bioingeniería (601E – Ingeniería de Tejidos) con sus respectivas condiciones de almacenamiento, riesgos asociados al manejo y rutas de descarte. Esta ventana permitió mediante botones el ingreso de nuevos residuos, edición, eliminación, generación de historiales y la visualización de notificaciones relacionadas con la permanencia de residuos en las zonas de almacenamiento del laboratorio, así como la generación del listado exportable de residuos. Además, este módulo incluyó una pestaña con información referente a la documentación legal y normatividad vigente para la gestión de residuos hospitalarios peligrosos y no peligrosos.

1.7. Gestión de Usuarios

Módulo que permitió el manejo de la información de los usuarios de la plataforma, cuyo acceso está permitido solo para los usuarios con rol de administrador. Mediante este módulo se pudieron crear nuevos usuarios, gestionar la asignación de roles estándar y administrador, edición y eliminación de los perfiles.

1.8. Líneas de investigación: biomecatrónica y biomateriales.

En esta etapa se replicó la plataforma para las diferentes líneas de investigación en el Laboratorio de Bioingeniería: biomecatrónica y biomateriales. Estas líneas de investigación incluyeron los mismos módulos desarrollados en los pasos anteriores, con bases de datos independientes.

2. Implementación de la infraestructura

Se realizó el montaje de la aplicación web en un servidor con sistema operativo Windows Server 2016 Desktop Experience, alojado en la intranet de la Universidad Pontificia Bolivariana, permitiendo el acceso de la plataforma desde todo el campus de la universidad y se le asignó el subdominio de <http://limbio.upb.edu.co> a la aplicación web.

3. Validación de los módulos

La validación de la aplicación web se basó en la usabilidad percibida por los clientes finales en el momento de navegar en la plataforma. Para esto, se realizó una encuesta basada en el Sistema de Escalas de Usabilidad [15] a 5 usuarios de la aplicación del Centro de Bioingeniería luego de hacer uso de esta, y se calculó el puntaje de la siguiente manera:

3.1. Puntuación de las preguntas pares: al resultado "X" se le debe calcular el valor de la siguiente manera:

$$5 - X$$

3.2. Puntuación de las preguntas impares: al resultado "X" se le debe calcular el valor de la siguiente manera:

$$X - 1$$

3.3. Puntuación final: luego de haber calculado el valor de cada pregunta se deben sumar todos y multiplicar el resultado por 2.5

4. Documentación del proyecto.

Se realizó un manual de usuario para facilitar el manejo de la plataforma, el cual incluye información sobre el acceso a esta, los roles de usuario y la navegación de la aplicación web. Además, se realizó una guía de

instalación del software en caso de que se necesite volver a instalar el servidor.

Resultados y análisis

Se desarrolló una aplicación web con tecnologías M.E.R.N. (MongoDB, Express, ReactJS, Node.JS) para el manejo integral de la información del Laboratorio de Bioingeniería para las tres líneas de investigación presentes: Ingeniería de Tejidos, Biomecatrónica y Biomateriales.

Se realizó el montaje de infraestructura en la intranet de la Universidad Pontificia Bolivariana, permitiendo el acceso a la plataforma, mediante un navegador web bajo la dirección del subdominio <http://limbio.upb.edu.co>, en todo el campus de la universidad, incluyendo las oficinas en el Centro de Bioingeniería y los computadores del Laboratorio de Bioingeniería.

Se generó la documentación de la aplicación, obteniendo así un Manual de Usuario y una Guía de instalación, este último se adjunta como anexo.

Los resultados de la encuesta se muestran en la tabla 1.

Tabla 1. Resultados de la encuesta de validación por usabilidad

Usuario	Puntaje [%]	Calificación
1	90	A
2	87.5	A
3	97.5	A
4	70	BB
5	65	CC
Promedio	82	

Como se observa en la tabla 1, hubo una puntuación promedio de **82% (CALIFICACIÓN A)**, esto quiere decir que la aplicación tiene percepción de usabilidad favorable por parte de los usuarios [16].

Por otra parte, el puntaje del usuario 5 está por fuera de la desviación estándar (12.38) con respecto al valor promedio, esto da indicio de un error de interpretación en el momento de responder la encuesta. Así, se encuentran preguntas como “*Me sentí algo incómodo al utilizar esta aplicación*” y “*Creo que necesitaría el soporte de un técnico para poder utilizar esta aplicación*” cuyas respuestas no coinciden con otras preguntas de la misma encuesta, tal como “*Creo que la aplicación es bastante fácil de utilizar*”.

Conclusiones

Se realizó el diseño y el desarrollo de la aplicación web LIMBIO como Sistema de Gestión de Información para el laboratorio de Bioingeniería, implementando una infraestructura en la intranet de la Universidad Pontificia

Bolivariana y el despliegue de la aplicación bajo el subdominio <http://limbio.upb.edu.co>.

Así mismo, se realizó la documentación de la plataforma con un manual de usuario y una guía de instalación, y, finalmente se validó la aplicación por usabilidad, dando como resultado una percepción favorable de la usabilidad de la aplicación LIMBIO.

Referencias Bibliográficas

- [1] Sapio Sciences, "What is a LIMS?," *Noviembre 28, 2016*. .
- [2] K. M. O 'Leary, "Selecting the Right LIMS Critiquing technological strengths and limitations," *Sci. Comput.*, 2012.
- [3] R. Fielding *et al.*, *Hypertext Transfer Protocol -- HTTP/1.1*. 1999.
- [4] M. S. Mikowski and J. C. Powell, *Single page web applications: JavaScript end-to-end*. Manning Publications, 2013.
- [5] LABVANTAGE Solutions Inc., "HOW DIFFERENCES IN TECHNOLOGY AFFECT LIMS FUNCTIONALITY , COST , & ROI," 2011.
- [6] H. S. Oluwatosin, "Client-Server Model," *IOSR J. Comput. Eng.*, vol. 16, no. 1, pp. 57–71, 2014.
- [7] D. Flanagan, *JavaScript: The Definitive Guide*, 6th ed. O'Reilly Media, Inc., 2011.
- [8] T. Kahlert and K. Giza, "Visual Studio Code: Tips & Tricks Vol. 1," vol. 1, no. March, M. Schiffer, Ed. Unterschleißheim: Microsoft Deutschland GmbH, 2018.
- [9] StackOverflow, "Developer Survey Results 2018," 2018. .
- [10] M. Al-Mukhtar and S. Hadi, "Developing a Three-Tier Web Data Management Application for Higher Education Admission Environment," *International Arab Journal of e-Technology*, vol. 2, no. June. pp. 175–180, 2012.
- [11] T. Khuat, "Developing a frontend application using ReactJS and Redux," Laurea University of Applied Sciences, 2018.
- [12] D. Schwarz, *Jump Start Adobe XD*. SitePoint, 2017.
- [13] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to build high-performance network programs," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 80–83, 2010.
- [14] L. Vokorokos, M. Uchnár, and A. Baláž, "MongoDB scheme analysis," *INES 2017 - IEEE 21st Int. Conf. Intell. Eng. Syst. Proc.*, vol. 2017–Janua, pp. 67–70, 2017.
- [15] "System Usability Scale (SUS) | Usability.gov." [Online]. Available: <https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html>. [Accessed: 26-Jul-2019].
- [16] MeasuringU, "MeasuringU: Measuring Usability with the System Usability Scale (SUS)." [Online]. Available: <https://measuringu.com/sus/>.

Anexos

Anexo 1: Guía de instalación de la aplicación.

Anexo 2: Encuesta de validación.

Anexo 3: Capturas de pantalla de la interfaz de la aplicación web LIMBIO.