

Modelo de curso para la asignatura de "Programación en la web" utilizando el framework de angular e implementación de un aplicativo web para un laboratorio clínico.

Jose Fabian Jimenez Ovalle y Javier Fernando Carvajal Sanabria

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Manuel Guillermo Florez Becerra

Msc. En informática

Codirector

Jathinson Meneses Mendoza

Msc. En Gestión, Aplicación y Desarrollo de Software.

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2023

### **Agradecimientos**

Primeramente, agradecido con Dios, la vida y el destino de permitirme estudiar Ingeniería de sistemas en la Universidad Industrial de Santander. En el transcurso de mi estadía conocí grandes profesores y mentores que me inspiraron para alcanzar mis objetivos, también tuve la oportunidad de compartir grandes experiencias con mis compañeros que pronto serán grandes profesionales.

De tal manera quiero expresar mi agradecimiento a mi núcleo familiar (padre y madre) quienes siempre estuvieron muy pendientes de mí me apoyaron demasiado al principio de la carrera donde fueron días y meses complicados para poder adaptarme a la etapa universitaria, dándome las fuerzas y palabras de apoyo para no rendirme.

También agradezco el acompañamiento en esta etapa final a los profesores Manuel Guillermo Flórez Becerra y Jathinson Meneses Mendoza quienes nos terminaron de guiar para cumplir con nuestro objetivo final, mil gracias por la confianza y paciencia.

Finalmente agradecer a mi compañero Jose Fabian Jiménez Ovalle por permitirme trabajar con él y ser su compañero en el trabajo de grado, por ser ese apoyo y darme fuerzas para poder cerrar esta etapa, por último, agradecer a mi compañero Jorge Sandoval quien me guio en gran parte de la carrera para poder alcanzar todos los objetivos.

***Javier Fernando Carvajal Sanabria.***

### **Agradecimientos**

Primero que todo, quiero agradecer a Dios la oportunidad de pertenecer a una universidad de tan alto nivel como es la Universidad Industrial de Santander. En todo el camino he vivido momentos de derrota, superación y desafío a lo largo de este camino, pero gracias a Dios y el apoyo incondicional de todas las personas que están a mi lado, he logrado un gran crecimiento tanto a nivel personal como profesional. Cada obstáculo lo tomé siempre como una oportunidad de aprendizaje y crecimiento, y cada logro fue celebrado con entusiasmo y la compañía de todos mis seres queridos.

Agradezco a mis padres, hermano, pareja, familia y amigos por el apoyo incondicional en esta magnífica experiencia. Su presencia en mi vida ha sido fortaleza, motivación e inspiración desempeñando un papel importante en este logro tan importante en mi vida.

Por último, pero no menos importante, quiero expresar mi gratitud al profesor Manuel Guillermo Florez Becerra y Jathinson Meneses Mendoza por su conocimiento, orientación y dedicación en la supervisión de este proyecto. Su experiencia, apoyo y consejos han sido un factor fundamental para la culminación de este trabajo y mi desarrollo profesional.

***Jose Fabian Jimenez Ovalle.***

## Contenidos

Introducción .....	13
1 Planteamiento y justificación del problema .....	15
2 Objetivos .....	17
2.1 Objetivo general.....	17
2.2 Objetivos específicos .....	17
3 Marco teórico .....	18
3.1 Marco de referencia conceptual .....	18
3.1.1 Desarrollo web .....	18
3.1.2 Aplicación web .....	19
3.1.2.1 Importancia de las aplicaciones web.....	19
3.1.2.2 Tipos de aplicaciones web .....	20
3.1.3 Arquitectura de software.....	21
3.1.3.1 importancia de la arquitectura.....	22
3.1.3.2 Patrones de arquitectura de software .....	22
3.1.4 Framework .....	25
3.1.5 Front-End .....	26
3.1.5.1 Angular .....	27
3.1.6 Back-End.....	27
3.1.6.1 NodeJS .....	28

3.1.6.2 Express .....	28
3.1.7 Bases de datos .....	29
3.1.7.1 Tipos de bases de datos .....	29
3.1.8 Aprendizaje en línea o E-learning.....	30
3.1.8.1 Plataforma LMS.....	32
3.1.9 Material Educativo Informático en la web.....	32
4 Estado del arte.....	32
4.1 Antecedentes de la situación de estudio.....	32
5 Metodología .....	34
5.1 Planning .....	34
5.2 Desarrollo.....	35
5.3 Review .....	36
6 Desarrollo de proyecto.....	36
6.1 Fase de planning .....	36
6.1.1 Herramientas .....	36
6.1.1.1 Stack de tecnologías.....	37
6.1.1.1.1 Angular .....	37
6.1.1.1.2 Bootstrap .....	37
6.1.1.1.3 PostgreSQL y Sequelize .....	37
6.1.1.1.4 NodeJS y Express .....	38

6.1.1.2 Control de tareas .....	38
6.1.1.2.1 Jira.....	38
6.1.1.3 Versionamiento .....	38
6.1.1.3.1 Git y Github .....	38
6.1.1.4 Testing.....	39
6.1.1.4.1 MochaJS.....	39
6.1.1.4.2 ChaiJS .....	40
6.1.1.5 Despliegue.....	40
6.1.1.5.1 Render .....	40
6.1.2 Análisis previo .....	40
6.1.2.1 Análisis de requerimientos.....	41
6.1.2.2 Requerimientos Funcionales.....	41
6.1.2.2 Requerimientos no Funcionales:.....	44
6.1.2.3 Casos de Uso.....	46
6.1.2.4 Diagramas de Actividades .....	47
6.1.2.5 Arquitectura .....	49
6.1.3 Diseño Interfaz de Usuario .....	50
6.2 Fase de desarrollo .....	51
6.2.1 Avance Sprints .....	51
6.2.2 Aplicación Web Laboratorio Clínico.....	59

6.2.2.1 Desarrollo Back-end .....	59
6.2.2.1 Desarrollo Front-end .....	64
6.2.3 Curso Programación Web .....	70
6.3 Fase de Review .....	75
6.3.1 Aplicación laboratorio clínico.....	75
6.3.1.1 Plan de pruebas .....	75
6.3.1 Aplicación laboratorio clínico.....	78
7 conclusiones .....	79
8 recomendaciones .....	80
Referencias bibliográficas.....	81

**Lista de tablas**

<b>Tabla 1.</b> Requerimiento funcional 1 .....	41
<b>Tabla 2.</b> Requerimiento funcional 2.....	42
<b>Tabla 3.</b> Requerimiento funcional 3.....	42
<b>Tabla 4.</b> Requerimiento funcional 4.....	42
<b>Tabla 5.</b> Requerimiento funcional 5.....	43
<b>Tabla 6.</b> Requerimiento funcional 6.....	43
<b>Tabla 7.</b> Requerimiento no funcional 1 .....	44
<b>Tabla 8.</b> Requerimiento no funcional 2.....	44
<b>Tabla 9.</b> Requerimiento no funcional 3.....	44
<b>Tabla 10.</b> Requerimiento no funcional 4.....	45
<b>Tabla 11.</b> Requerimiento no funcional 5.....	45
<b>Tabla 12.</b> Requerimiento no funcional 6.....	45
<b>Tabla 13.</b> Plan de pruebas .....	75



## Lista de figuras

<b>Figura 1.</b> Patrón cliente-servidor .....	23
<b>Figura 2.</b> Patrón de capas .....	23
<b>Figura 3.</b> Patrón maestro-esclavo.....	24
<b>Figura 4.</b> Patrón modelo-vista-controlador .....	25
<b>Figura 5.</b> Actores del Sistema .....	41
<b>Figura 6.</b> Diagrama de casos de uso .....	46
<b>Figura 7.</b> Proceso para iniciar sesión, es el mismo para cada rol de usuario .....	47
<b>Figura 8.</b> Registro creación de cuenta.....	47
<b>Figura 9.</b> Ver resultados .....	48
<b>Figura 10.</b> Subir resultados .....	49
<b>Figura 11.</b> Arquitectura.....	50
<b>Figura 12.</b> Mockup de interfaz para iniciar sesión.....	51
<b>Figura 13.</b> Tareas sprint 1 .....	52
<b>Figura 14.</b> Tareas sprint 2 .....	52
<b>Figura 15.</b> Tareas sprint 3 .....	53
<b>Figura 16.</b> Tareas sprint 4 .....	53
<b>Figura 17.</b> Tareas sprint 5 .....	54
<b>Figura 18.</b> Tareas sprint 6 .....	54
<b>Figura 19.</b> Tareas sprint 7 .....	55
<b>Figura 20.</b> Tareas sprint 8 .....	55
<b>Figura 21.</b> Tareas sprint 9 .....	56
<b>Figura 22.</b> Tareas sprint 10 .....	57

<b>Figura 23.</b> Tareas sprint 11 .....	57
<b>Figura 24.</b> Tareas sprint 12 .....	58
<b>Figura 25.</b> Tareas sprint 13 .....	58
<b>Figura 26.</b> Diagrama de flujo acumulado .....	59
<b>Figura 27.</b> Arquitectura API Rest .....	60
<b>Figura 28.</b> Estructura Back-end .....	60
<b>Figura 29.</b> Estructura de Carpetas Back-end .....	61
<b>Figura 30.</b> Estructura Carpetas Inicial Front-end.....	64
<b>Figura 31.</b> Estructura Front-end laboratorio .....	66
<b>Figura 32.</b> Vista Inicial home .....	67
<b>Figura 33.</b> Vista de servicios.....	68
<b>Figura 34.</b> Vista de resultados por paciente .....	68
<b>Figura 35.</b> Vista inicio de sesion.....	69
<b>Figura 36.</b> Vista registro de usuario.....	69
<b>Figura 37.</b> Lista de tareas para el curso de programación web.....	70
<b>Figura 38.</b> Planificación Curso programación en la web con Angular en Meiweb .....	71
<b>Figura 39.</b> Planificación Curso programación en la web con Angular en Moodle.....	72
<b>Figura 40.</b> Contenido curso programación en la web .....	73
<b>Figura 41.</b> Hoja de ruta para las clases de cada semana .....	74

### Resumen

**Título:** Modelo de curso para la asignatura de "Programación en la web" utilizando el framework de angular e implementación de un aplicativo web para un laboratorio clínico.\*

**Autores:** Javier Fernando Carvajal Sanabria y Jose Fabian Jimenez Ovalle\*\*

**Palabras clave:** Programación web, E-learning, Framework, Angular, NodeJS, Express

**Descripción:** Se tratan algunos temas importantes para el desarrollo del proyecto, tales como: términos de programación en la web, introducción a nuevas tecnologías para el desarrollo web y demás conceptos que faciliten el entendimiento y la realización del proyecto, lo anterior, basándose en fundamentos teóricos y contenidos similares a lo planteado. En busca de lo anterior se formaliza una hoja de ruta, basada en el planteamiento y justificación del problema, los objetivos proyectados, el cronograma, la metodología y el presupuesto necesario para la ejecución del proyecto, adicionalmente se muestra el desarrollo del proyecto mostrando los requerimientos, diagramas de actividades, arquitectura del proyecto etc. Finalmente se describen las pruebas que se realizaron del proyecto, mostrando la funcionalidad de la página y el contenido de los cursos desarrollados para las plataformas de aprendizaje Moodle y MeiWeb, luego se deja una conclusión basada en los objetivos generales y específicos descritos al principio del documento, para finalizar se dan recomendaciones para trabajos futuros y mejoras que se podrían realizar al proyecto para que de esta manera genere un mayor impacto para los estudiantes en la asignatura programación en la web.

---

\* Proyecto de Grado. Trabajo de Investigación.

\*\* Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Msc. En informática Manuel Guillermo Florez Becerra

Codirector: Msc. En Gestión, Aplicación y Desarrollo de Software Jathinson Meneses Mendoza

### Abstract

**Title:** Course Model for the Subject "Web Programming" using the Angular Framework and Implementation of a Web Application for a Clinical Laboratory.\*

**Authors:** Javier Fernando Carvajal Sanabria and Jose Fabian Jimenez Ovalle\*\*

**Keywords:** Web Programming, E-learning, Framework, Angular, NodeJS, Express.

**Description:** Various important topics for the project's development are addressed, such as web programming terminology, an introduction to new technologies for web development, and other concepts that facilitate the understanding and execution of the project. This is done based on theoretical foundations and content similar to what was proposed. In pursuit of these objectives, a roadmap is formalized, based on the problem statement and justification, projected objectives, schedule, methodology, and the budget necessary for project execution. Additionally, the project's development is presented, including the requirements, activity diagrams, project architecture, etc. Finally, the document describes the project's testing, showcasing the website's functionality and the content of the courses developed for the Moodle and MeiWeb learning platforms. Subsequently, a conclusion is drawn based on the general and specific objectives outlined at the beginning of the document. To conclude, recommendations for future work and potential project enhancements are provided, with the aim of generating a greater impact on students in the web programming subject.

---

\* Degree Project. Research Work.

\*\* Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering.

Director: Msc. En informática Manuel Guillermo Florez Becerra

Co-Director: Msc. En Gestión, Aplicación y Desarrollo de Software Jathinson Meneses Mendoza

## **Introducción**

Actualmente, se debe intentar obtener los mejores beneficios posibles del internet, por esta razón, las aplicaciones web se han convertido en una herramienta esencial para las empresas y comercios. Además de permitir ofrecer sus productos a más clientes y automatizar procesos de negocio, una página web es la imagen de presentación ante los usuarios de la marca, por lo que influenciará en gran medida al usuario en su decisión de hacer o no una compra.

Aunque existen varios lenguajes de programación dedicados al desarrollo web, hoy en día, Javascript es uno de los lenguajes más utilizados por los desarrolladores por su flexibilidad y todas las funcionalidades que lo hacen un lenguaje muy potente. A medida que crece su popularidad, cada vez se dan a conocer nuevas tecnologías que facilitan el trabajo de desarrollo y ofrecen nuevas capacidades al desarrollador. Angular y NodeJS son algunas de estas tecnologías y nos brindan la posibilidad de hacer todo el desarrollo, front-end y back-end, con JavaScript.

Con este proyecto, los estudiantes de ingeniería de sistemas que se encuentren cursando la asignatura de programación en la web, podrán encontrar un curso de aprendizaje para Angular y NodeJS en la plataforma de la asignatura, MeiWeb. En el curso se desarrolla un prototipo de aplicación web para la muestra de resultados de laboratorio clínico.

El proyecto se trabajó con una metodología basada en el marco de trabajo SCRUM. Inicialmente, está la fase de planning donde se analizaron los requerimientos y, en base en esto, se planificaron las siguientes etapas. Luego, viene la fase de desarrollo en la cual se buscó la ejecución del prototipo. Finalmente, en la fase de review por medio de retroalimentación se valoró el avance logrado.

Antes de cada sprint se hizo una planeación de lo que se quiere alcanzar, durante el sprint hubo espacios periódicos con el fin de comentar los avances y finalmente está el review, donde se valoró el avance logrado.

## **1 Planteamiento y justificación del problema**

En los últimos años hemos podido observar la evolución que ha tenido el mundo, el internet, los smartphones y computadores portátiles hacen el mundo cada vez más pequeño y nos ofrecen la posibilidad de tener las cosas a unos cuantos clics de distancia. Esto ha provocado que las personas, empresas e industrias cambien su forma de actuar con el fin de poder aprovechar todas las ventajas que ofrece la tecnología.

Si hablamos de educación, cada vez es más común encontrar que los estudiantes tienen más medios de acceder a recursos virtuales que le ofrecen flexibilidad de tiempo y ubicación, ya que el usuario puede acceder al material del curso a la hora y lugar donde más cómodo sea posible. Incluso, a menudo se encuentran programas que no ofrecen el servicio presencial por lo que por medio de estos cursos online tienen una oportunidad que no tienen por medio de la modalidad presencial. En nuestro paso por la universidad cursamos diferentes materias y una que nos llamó la atención fue programación en la web, ya que nos brindaba la oportunidad de desarrollar un perfil competitivo en el mercado.

Por esta razón, para este proyecto se va a desarrollar un curso de aprendizaje complementario para la materia programación en la web con el fin de incentivar a los estudiantes al aprendizaje de nuevas tecnologías por medio de un aplicativo web, este curso se divide por semanas donde en cada una de ellas se estudiarán los temas relacionados con la materia, en cada guía de la semana se tendrá información y ejemplos los cuales ayudarán al estudiante a cumplir con el ejercicio que se deje al final de la clase, mientras las semanas pasan los estudiantes irán construyendo una página web la cual busca ayudar en la gestión de muestra de resultados a los pacientes y a los encargados de un laboratorio clínico, ya que el paciente tendría acceso a los resultados de sus exámenes con tan solo ingresar al sitio web del laboratorio e identificarse con las

credenciales de acceso. Al final del curso el estudiante tendrá la oportunidad de presentar su proyecto final en base a lo construido durante todas las semanas del semestre, el estudiante podrá decidir si continuar con el proyecto propuesto en clase agregándole nuevas funcionalidades o iniciar un proyecto con una propuesta diferente que cubra los temas vistos durante el semestre.



## **2 Objetivos**

### **2.1 Objetivo general**

Diseñar e implementar un curso de aprendizaje complementario para la asignatura de "Programación en la web" utilizando un prototipo con el Stack de tecnologías Angular y NodeJS utilizando las plataformas de enseñanza virtual MeiWeb y Moodle.

### **2.2 Objetivos específicos**

- Diseñar un curso de aprendizaje virtual en la plataforma MeiWeb y el aula virtual de aprendizaje Moodle, como complemento para los estudiantes en la asignatura programación en la web (pgweb).
- Implementar un prototipo de aplicación web para la muestra de resultados de exámenes en laboratorios clínicos y otras funcionalidades de uso generalizado.
- Utilizar el aplicativo implementado como una fuente de ejemplos ilustrativos e integrarlos al curso de aprendizaje para programación en la web.
- Desarrollar competencias para incentivar el pensamiento creativo de los estudiantes, a partir del aprendizaje basado en competencias prácticas aplicando la teoría del curso implementado.

### **3 Marco teórico**

#### **3.1 Marco de referencia conceptual**

##### ***3.1.1 Desarrollo web***

El desarrollo web es una herramienta que cada vez cobra más importancia para los negocios y empresas, anteriormente, no era muy común que una empresa tuviera una aplicación web, en cambio, hoy si una empresa no tiene página web está perdiendo mucho terreno en comparación a la competencia.

El desarrollo web es el proceso de creación y mantenimiento de sitios web. Puede implicar una amplia gama de acciones, desde la codificación y el diseño hasta la gestión de contenidos y la administración del servidor web (Tekla, 2022).

Los desarrolladores web forman parte de una de las disciplinas con más demanda en el mercado laboral actualmente. Esto provoca que las capacitaciones y formaciones en desarrollo web sean muy apetecidas por estudiantes y profesionales que desean involucrarse en esta área.

En todo el proceso de desarrollar una aplicación web se pueden emplear varios lenguajes, cada uno con sus características que hacen que destaque sobre otros según su caso de uso. HTML, CSS y Javascript son los lenguajes base para el desarrollo web. HTML es un lenguaje de marcado de hipertexto y su función principal es otorgar una estructura a la página web, CSS es un lenguaje de diseño y se encarga de las vistas y presentación del sitio web, por último, Javascript es un lenguaje de programación interpretado y tiene la tarea de añadir funcionalidades a la página para que sea interactiva (Fantino, 2021).

### ***3.1.2 Aplicación web***

Una aplicación web es un programa de aplicación que se almacena en un servidor remoto y se entrega a través de Internet a través de una interfaz de navegador. (TechTarget) Una de las ventajas de esta aplicación es que no tienen la necesidad de instalar software para su mantenimiento y actualización por lo que no consume recursos del equipo ya que es ejecutada sobre un navegador web y soportada en uno o más servidores.

Una aplicación web puede ser accedida por sus usuarios en un navegador web a través de un enlace, solo hace falta internet para que los usuarios puedan entrar en cualquier lugar donde se encuentren. A diferencia de una página web, que contiene información estática, las aplicaciones web son un grupo de páginas dinámicas que varían su contenido según la interacción del usuario.

**3.1.2.1 Importancia de las aplicaciones web.** Si se trata del tema comercial, las aplicaciones web se han convertido en un recurso obligatorio si se quiere luchar con los competidores, ya que, como se ha dicho anteriormente, permiten acercarse más al usuario y otorgar un servicio más personalizado. Las empresas que no se adaptan a este nuevo marco pierden ventaja competitiva y se quedan estancadas en el mercado hasta llegar al punto de fracasar y desaparecer. Una gran cantidad de recursos y procesos pueden ser optimizados e incluso automatizados.

También es posible utilizar técnicas de recolección de datos por medio de estas aplicaciones con el fin apoyar la toma de decisiones de la empresa (Tiuso, 2020b).

**3.1.2.2 Tipos de aplicaciones web.** A continuación, se encuentra información sobre tipos de aplicaciones web

- Aplicación web estática

Son aplicaciones con poca información y no suelen presentar muchos cambios, por lo tanto, se muestra contenido fijo para el usuario, aunque puede tener contenido dinámico como imágenes, gif o vídeos. El proceso de modificación o actualización de estas aplicaciones es un poco complicado, ya que es necesario editar el HTML y volver a subirlo al servidor donde se encuentre alojada la aplicación. Un portafolio o curriculum digital son ejemplos de este tipo de aplicación (YeePLY).

- Aplicación web dinámica

Son más complejas a nivel técnico, ya que a cada usuario se le carga información distinta según su perfil, esta información se carga desde bases de datos. Para el desarrollo se suelen utilizar lenguajes como JavaScript, PHP, entre otras. El proceso de actualización es más sencillo que el de las aplicaciones estáticas. un ejemplo muy común son los foros (YeePLY).

- E-commerce o tienda virtual

Son aplicaciones pensadas para negocios con ventas electrónicas, su desarrollo es complicado porque deben crearse unas pasarelas de pago para tarjetas de crédito, PayPal, etc. También debe permitir la gestión del negocio a un rol de administrador que se encargará de actualizar inventario, gestionar precios y pedidos. Amazon y Mercado Libre son grandes ejemplos (YeePLY).

- Portal web app

Un portal web es un tipo de aplicación que ofrece al usuario el acceso a varios recursos y servicios sobre una misma aplicación. Este portal puede contener foros, documentos, tienda electrónica, etc. (Tokio school).

- Aplicación web con gestor de contenidos

Estas aplicaciones se utilizan para los casos en que el contenido se debe actualizar o ampliar frecuentemente. Tienen un gestor de contenidos (CMS) a través del cual el administrador puede ir realizando los cambios y actualizaciones él mismo. Periódicos digitales y medios de comunicación son ejemplos de una aplicación web con gestor de contenido (YeePLY).

### ***3.1.3 Arquitectura de software***

La arquitectura de software es un concepto que se refiere a una planificación basada en modelos, patrones y abstracciones teóricas, a la hora de la construcción de cualquier software de cierta complejidad y como paso previo a cualquier implementación. La arquitectura se refiere al diseño y planificación a un nivel conceptual superior de un sistema (Calle, 2020). Esta planificación permite disponer de una guía teórica que facilita el desarrollo con un mejor conocimiento de las reglas que hay que seguir para lograr el producto deseado. El término patrón se refiere a una solución que se pueda reutilizar más adelante en proyectos o casos similares. La arquitectura de software se basa en los requerimientos del cliente, por lo que es primordial realizar un análisis detallado para definir que se quiere hacer y cómo se va a hacer. El diseño de la arquitectura no es algo definitivo ya que puede sufrir modificaciones durante el desarrollo con el fin de buscar el mejor entregable.

Los requerimientos de usuario ayudan a visualizar el panorama del logro que se quiere alcanzar y así definir las herramientas que mejor se adaptan a la solución planteada. En el desarrollo de software existen muchas tecnologías y cada una de ellas ofrece recursos que se adaptan mejor en dado caso. El diseño de arquitectura puede llegar a variar en algunas partes según las tecnologías y frameworks usados.

**3.1.3.1 importancia de la arquitectura.** Es de suma importancia planificar detalladamente y elegir minuciosamente las herramientas y tecnologías con las que se desarrollará el proyecto. Si se estructura una buena arquitectura se pueden optimizar recursos humanos y económicos, esto debido a que el desarrollo es más eficiente.

La arquitectura de software es importante porque tiene la responsabilidad de brindar calidad y escalabilidad a las aplicaciones. Es decir, la arquitectura de software tiene el objetivo de mejorar el desempeño, la usabilidad, el ahorro de tiempo y la capacidad del software a recibir modificaciones que se acoplen eficientemente de forma que sean mantenibles y testeables (Huet, 2022).

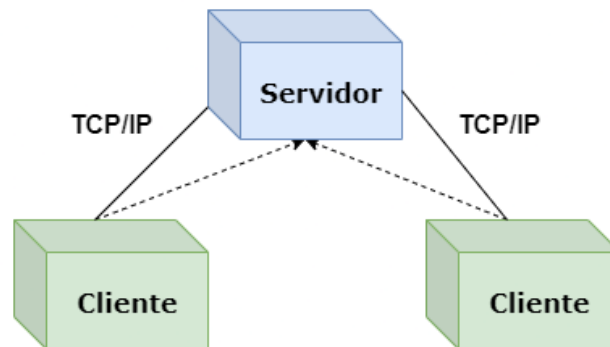
**3.1.3.2 Patrones de arquitectura de software.** Patrones más conocidos.

- Patrón cliente-servidor

Consiste en 2 partes fundamentales, un servidor y una serie de clientes. Los clientes mandan una petición al servidor y este envía una respuesta a dicha solicitud. La administración de esta arquitectura no tiene mucha complejidad ya que todo se encuentra en un servidor centralizado, esto puede llegar a provocar problemas de disponibilidad. El e-mail y documentos de uso compartido son aplicaciones que trabajan con esta arquitectura.

**Figura 1.**

*Patrón cliente-servidor*



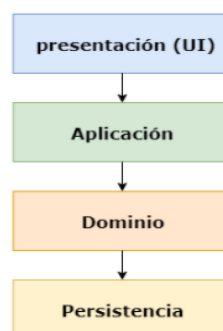
Adaptado de <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>

- Patrón de capas

Cuando se presenta una aplicación que se puede descomponer en capas, cada capa es una subtarea, el patrón de capas es el más adecuado. cada se encuentra en un nivel de abstracción diferente y se encarga de proporcionar servicios a la siguiente capa de mayor nivel (Huet, 2022). Las aplicaciones de e-commerce usan este patrón de arquitectura de capas.

**Figura 2.**

*Patrón de capas*

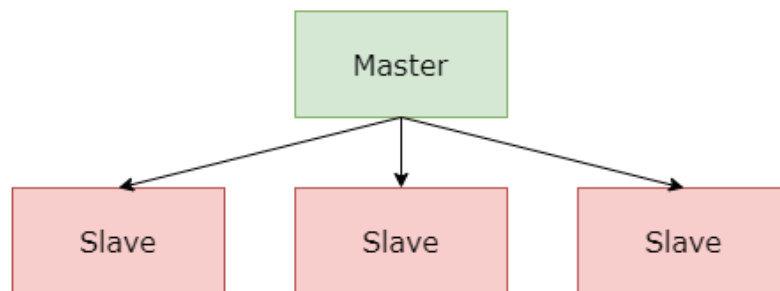


- Patrón maestro-esclavo

Maestro-esclavo es un patrón que se divide en 2 grupos, como su nombre lo indica, está el componente maestro que distribuye las tareas entre los componentes esclavos, estos le devuelven resultados del trabajo propuesto para que el maestro los procese y almacene.

**Figura 3.**

*Patrón maestro-esclavo*



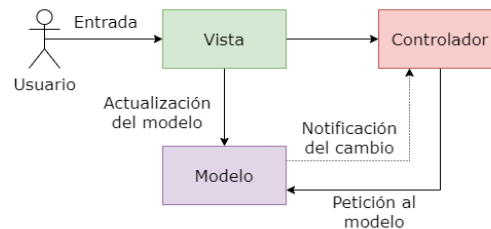
- Patrón modelo-vista-controlador

También conocido como MVC, este patrón consiste en 3 componentes. Está el modelo, el componente principal, donde se encuentran los datos de la aplicación. La vista está diseñada para mostrar la información al usuario, una aplicación puede tener múltiples vistas. Por último, el controlador maneja la entrada del usuario (Huet, 2022). Frameworks como Angular en Javascript, Spring en Java y Django en Python se basan en este patrón.



**Figura 4.**

*Patrón modelo-vista-controlador*



#### - Patrón de arquitectura de microservicios

La arquitectura de microservicios está diseñada para dividir un programa grande en numerosos programas pequeños que cumplen una labor en específico. Cada microservicio se ejecuta libremente de forma independiente y puede estar en el lenguaje que mejor se adapta a la función que tiene que cumplir (Calle, 2020).

#### **3.1.4 Framework**

Un Framework es una estructura base utilizada como punto de partida para elaborar un proyecto con objetivos específicos. Es una especie de plantilla, un esquema conceptual, que simplifica la elaboración de una tarea (Muyente, 2020). Un Framework es un modelo que sirve como base para que el desarrollo sea mucho más ágil y eficiente, ya que solo es necesario complementar y adaptar ese modelo de acuerdo con lo que se quiere lograr.

En el desarrollo de un proyecto, iniciar desde cero puede significar un gran gasto de recursos. En cambio, si utilizamos algún framework, el desarrollador podrá optimizar tiempo para dedicar en otras funcionalidades del proyecto, así mismo, se reducirán errores y entregará un código más limpio, reutilizable y con menos repetición.

Existe una gran variedad de frameworks con los que se puede trabajar hoy en día, pero hay algunos con más popularidad que otros, esto significa que tienen una comunidad más extendida por lo que resulta más sencillo encontrar documentación, información y módulos ya desarrollados listo para su uso.

### ***3.1.5 Front-End***

Un sistema de *front-end* es parte de un sistema de información al que el usuario accede directamente e interactúa para recibir o utilizar las capacidades de back-end del sistema anfitrión. Permite a los usuarios acceder y solicitar las prestaciones y servicios del sistema de información subyacente (Urruttia 2021).

Cuando hablamos de Front-End hacemos referencia a la parte visual de nuestra aplicación web, la cual está compuesta por una estructura, colores, imágenes entre otras características, desde aquí el cliente podrá interactuar con la aplicación de manera interactiva.

Una persona que opta por ser desarrollador Front-End debe manejar varias tecnologías principalmente HTML, Css y JavaScript, también debe tener nociones de diseño UX y UI.

- HTML es un lenguaje de hipertexto el cual se encarga de la estructura de nuestra página web por medio de etiquetas.
- Css son hojas de estilos las cuales tienen como función agregar diferentes propiedades, con el fin de mejorar la parte visual de nuestra página.
- JavaScript es el lenguaje de programación que acompaña el desarrollo de las páginas web, su función es interactuar con el usuario.

- Diseño UX hace referencia a la experiencia del usuario, Diseño UI hace referencia a la interfaz.

**3.1.5.1 Angular.** Angular es un framework open source desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application) (Quality Devs, 2019). Este framework está desarrollado en typescript, es utilizado para versiones de escritorio como de móviles, nos sirve para modular nuestro código por medio de componentes, servicios e interfaces. En el transcurso del tiempo Angular ha tenido diferentes versiones en este momento se encuentra en la versión 12, Angular es un framework muy popular ya que opta por unas características bastantes eficaces y amigables con el desarrollador como la velocidad de generación de código, su división de código hace más entendible el desarrollo, también contamos con angular CLI el cual nos ayuda a crear componentes, módulos y servicios, por medio de la consola de manera más rápida.

### ***3.1.6 Back-End***

El Back-End es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione. Consiste en el conjunto de acciones que pasan dentro de una web, pero que no podemos ver. Un ejemplo de esto es la comunicación con el servidor (Nestrategia). También hace referencia a toda la arquitectura del sitio como tal, está pendiente de que toda la lógica del negocio funcione correctamente.

A diferencia del Front-End, en el Back-end no se interactúa con el cliente de manera gráfica, ya que todas las funcionalidades como la seguridad y optimización no son visibles para los usuarios. Desde el Back-end también se gestionan las conexiones a bases de datos y la configuración de servidores.

**3.1.6.1 NodeJS.** Node.js es un entorno de tiempo de ejecución de JavaScript (de ahí su terminación en .js haciendo alusión al lenguaje JavaScript). Este entorno de tiempo de ejecución en tiempo real incluye todo lo que se necesita para ejecutar un programa escrito en JavaScript (Lucas, 2019).

Con Node.js podemos diseñar sitios web totalmente escalables, muy rápidos para la ejecución del código, no tendremos problemas de búfer ya que los datos se generan en chunks y nunca llegan al búfer, también es totalmente asíncrono y controlado por eventos por lo tanto no se tendrán bloqueos.

Gracias a todas estas características de Node.js grandes marcas han optado por utilizarlo como Microsoft, Netflix, Uber entre otras, ya que otorga flexibilidad en aplicaciones de tipo streaming, pago único, vinculadas E/S e intensivas de datos en tiempo real.

**3.1.6.2 Express.** Express es un framework o marco de trabajo desarrollado sobre Node.js, lo que quiere decir que provee toda la funcionalidad que necesitas para hacer aplicaciones web, permitiendo estructurar tu proyecto de una manera más fácil (Rodas, 2018).

Con express podemos manejar las peticiones HTTP en cualquier ruta, también nos sirve para lograr renderizar las vistas y poder emitir datos, con los middleware se añadirá procesamiento de peticiones, este framework es muy flexible y ahorra mucho código que se haría solo con Node.js, esta herramienta es fundamental en el desarrollo Back-End, ya que nos brinda visibilidad y escalabilidad en el código que se está desarrollando.

### ***3.1.7 Bases de datos***

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático (Oracle). La gestión de la base de datos se controla mediante sistemas de gestión, también llamados DBMS, este sistema está diseñado para el almacenamiento organizado y la extracción de los datos. Las bases de datos evitan que exista redundancia y garantizan integridad e independencia lógica de los datos.

**3.1.7.1 Tipos de bases de datos.** Según la necesidad, la organización y el uso que se le quieran dar a los datos se puede escoger entre los muchos tipos de bases de datos que existen, cada una tiene su modo de organización y propósito bien definido, por lo que es importante definir dónde se van a guardar los datos de la aplicación.

- Base de datos relacional

Este tipo de base de datos almacena los datos en tablas con filas y columnas. Los campos en las filas se conocen como registros y las columnas como atributos. Cada tabla tiene un atributo único que tiene el nombre de primary key y utilizado como el identificador de la tabla. La relación entre tablas es posible por medio de la primary key que direcciona al foreign de la otra tabla.

El SQL (lenguaje de consulta estructurada) es un lenguaje de programación que utilizan casi todas las bases de datos relacionales para consultar, manipular y definir los datos, además de para proporcionar control de acceso (Oracle).

- Base de datos distribuida

Son bases de datos con varios archivos que se encuentran distribuidos que se encuentran distribuidos en distintos lugares. La distribución de los datos asegura una mejor disponibilidad de los datos, pero puede generar duplicidad y riesgos de seguridad.

- Base de datos orientada a objetos

La información se almacena siguiendo el paradigma de la programación orientada a objetos, representando los datos en objetos y clases y la capacidad de encapsulamiento de los objetos.

- Base de datos NoSQL

Las bases de datos no relacionales tienen este nombre porque, por lo general, no utilizan el SQL y ofrecen la flexibilidad de almacenar datos tanto semiestructurados como estructurados. Las bases de datos NoSQL son de las más populares en las aplicaciones web debido al gran volumen de datos que puede almacenar ya que representan un gran ahorro de espacio comparado con las bases de datos relacionales. MongoDB, Cassandra, Neo4J y Redis son algunas de las bases de datos NoSQL más reconocidas en la actualidad.

Todas estas bases de datos tienen en común que no utilizan un modelo relacional. Además, tienden a ser más específicos en el tipo de datos que admiten y en cómo se pueden consultar los datos (Microsoft). Por ejemplo, las bases de datos documentadas son útiles cuando se desea guardar datos que no tienen una estructura fija y las bases de datos de serie temporal están optimizadas para realizar consultas basadas en una secuencia de tiempo.

### ***3.1.8 Aprendizaje en línea o E-learning***

Este concepto hace referencia a todas las actividades formativas que se dan exclusivamente a través de un dispositivo conectado a la red, lo que se suele llamar como aprendizaje electrónico, tele formación, formación online o aprendizaje virtual (Santander Universidades, 2022). También es conocida como formación a distancia o formación online. Este proceso de formación sucede en

la virtualidad por lo que el estudiante tiene la ventaja y la responsabilidad de autogestionar su proceso de aprendizaje. El E-learning se está haciendo cada vez más popular, es una de las modalidades de aprendizaje más usadas actualmente ya que ofrece una flexibilidad a los estudiantes que ninguna otra modalidad lo hace, la accesibilidad a la educación es mayor ya que dejan de existir barreras temporales y geográficas, existe una mayor oferta académica ya que muchos de los cursos en modalidad virtual no tienen una oferta presencial, los cursos están disponibles todo el tiempo y no hay necesidad de acoplarse a los calendarios de las instituciones académicas.

En Colombia, E-learning se ha convertido en una herramienta de ahorro para las empresas en sus costos de capacitación liberando su equipo humano usando una plataforma educativa digital. (Virtual Technologies). Esta modalidad permite ahorrar gastos en transportes, espacios físicos y ajustes de calendario de los empleados. El empleado tiene la oportunidad de adquirir conocimiento a su ritmo y en los temas que más le interesen, lo que provoca una mayor motivación y mejor rendimiento en sus actividades.

Los cambios y oportunidades que trajo consigo la pandemia del COVID-19 ha provocado que el aprendizaje en línea tome muchas más relevancia en el mercado educativo. el E-learning proyecta ganancias en Latinoamérica de más de 3 billones de dólares para el año 2023 (Fit Learning). Esto significa que en Colombia la modalidad de aprendizaje virtual ya es parte de la formación académica de miles de personas, ya sea como un complemento en su formación presencial o completamente su formación.

**3.1.8.1 Plataforma LMS.** Una plataforma LMS (Learning Management System) o en español, Sistema para la Gestión del Aprendizaje, es una herramienta de apoyo para el proceso de enseñanza-aprendizaje a distancia o semipresencial (Munte, 2020). LMS es la plataforma más utilizada para E-learning y hace posible la creación, almacenamiento, distribución y gestión de recursos para una clase en ambiente virtual. Estas plataformas están formadas por un servidor y una interfaz de usuario que es gestionada por instructores para el uso de los estudiantes.

### ***3.1.9 Material Educativo Informático en la web***

Más conocido como Meiweb por sus siglas en español, es una plataforma de aprendizaje virtual que hace posible la interacción y comunicaciones entre profesor y sus alumnos. El Meiweb es utilizado en la Universidad Industrial de Santander como un recurso para las asignaturas de programación en la web y sistemas operacionales en las que se gestiona la asistencia, el material de contenido del curso, calendario, actividades y evaluaciones programadas.

## **4 Estado del arte**

### **4.1 Antecedentes de la situación de estudio**

Para este ítem del documento se tratarán algunos antecedentes y trabajos del estado del arte que se encuentren orientados hacia el e-learning y plataformas de aprendizaje virtual y laboratorios clínicos que gestionen toda la muestra de resultados por medio de aplicaciones web. Actualmente, existen muchas plataformas dedicadas al aprendizaje virtual, pero en este caso, hablaremos puntualmente de 3 que son las que más popularidad tienen.

- Platzi: Es una plataforma enfocada principalmente al aprendizaje en temas de tecnología como programación, arquitectura de software y ciencia de datos, pero



también ofrece cursos de finanzas, idiomas, entre otros. Es usada por estudiantes y profesionales con el fin de afianzar sus conocimientos y crecer en su vida profesional.

- Udemý: Tiene disponible más de 180000 cursos y cuenta con más de 44 millones de estudiantes. Se define a sí misma como una tienda de aprendizaje virtual que tiene como meta mejorar las vidas de las personas por medio del aprendizaje.

- KhanAcademy: es una aplicación en la cual el estudiante puede obtener videos instructivos, ejercicios de práctica de forma virtual. Es usada por profesores como un complemento extra-clase para sus estudiantes.

- Moodle: Es un LMS implementado por varias instituciones educativas en Colombia, se encuentra disponible en varios idiomas y ofrece la posibilidad de crear ambientes de estudio personalizados.

ahora se tratarán y se propondrán como ejemplo algunos laboratorios clínicos que hacen todo el proceso de muestra de resultados por medio de aplicaciones web.

- SYNLAB: Laboratorio con más presencia en Europa y que ofrece sus servicios en más de 40 países. Tienen un portafolio de aproximadamente 5000 pruebas y procesan diariamente más de 1500 resultados clínicos.

- Laboratorio clínico de especialidades Bolívar: es un laboratorio clínico fundado en Bucaramanga encargado de la toma de exámenes para diagnóstico médico. Cuenta con reconocimientos nacionales e internacionales por servicio de calidad.

- Higuera Escalante: es un laboratorio clínico y banco de sangre altamente certificado. Tiene un alto reconocimiento en el oriente Colombiano. Además, Higuera Escalante tiene un área de investigación que tiene como objetivo generar conocimiento para una mejor práctica y diagnóstico en los examen clínicos.

## **5 Metodología**

Para el desarrollo de este proyecto los autores trabajan con una metodología basada en el marco de trabajo Scrum metodología donde se definen y aplican un conjunto de buenas prácticas con el fin de ser eficaces. Scrum es bastante importante para los proyectos donde los requisitos no están bien definidos y que exigen resultados a corto plazo. En conclusión, se utilizará Scrum porque el equipo pretende desarrollar un proyecto de forma ágil, pero a su vez, ahorrando tiempo y recursos.

Con base en lo anterior, el presente proyecto consta de 3 etapas que se consideran fundamentales para el alcance de los objetivos planteados en un corto plazo. A continuación, se profundizará un poco más de cada fase.

### **5.1 Planning**

Para la fase de planning se realiza una investigación sobre el foco del proyecto, en este caso, laboratorios clínicos que ayude a entender un poco más el problema y la solución planteada, además se revisarán proyectos de aplicaciones web desarrollados con las tecnologías definidas, Angular y NodeJS, con el fin de ejercer buenas prácticas en la implementación de la aplicación. Finalmente, se planifica el backlog del proyecto y se programan las tareas en los sprints.

- Investigar sobre las nuevas tecnologías usadas en el desarrollo web y cómo se trabajarán en este proyecto.
- Realizar diagrama de actividades que apoyen la gestión y control en la implementación del proyecto.

- Definir y diseñar el contenido para el modelo de curso para programación en la web utilizando el framework de angular e implementación de un aplicativo web para la muestra de resultados de laboratorio clínico.
- Definir el backlog de temas que se llevarán a cabo durante todo el proceso de desarrollo del proyecto.

## **5.2 Desarrollo**

Para esta etapa se desarrollan los sprints propuestos en el planning, con el fin de llevar a cabo el prototipo para la muestra de laboratorios clínicos y de esta manera complementarlo con el curso de aprendizaje en línea.

- Elaborar documentos teóricos sobre los temas de las tecnologías que vamos a usar para el desarrollo del prototipo.
- Proponer ejercicios para que los estudiantes realicen y puedan afianzar sus conocimiento vistos en clase.
- Desarrollar el prototipo para la muestra de resultados de laboratorio clínico, con el fin de complementarlo con documentación que sirva para enriquecer el curso que se está elaborando paralelamente.
- Subir el curso para la asignatura de "Programación en la web" en la plataforma MeiWeb para el uso de los estudiantes.

### **5.3 Review**

Finalmente se aplican diferentes pruebas al prototipo para comprobar su buen funcionamiento y su eficacia, con el propósito de realizar mejoras en un futuro para obtener una página más profesional.

## **6 Desarrollo de proyecto**

El desarrollo del proyecto se llevó a cabo en 3 etapas principales según como se mencionó en la metodología establecida.

### **6.1 Fase de planning**

#### ***6.1.1 Herramientas***

Esta sección detalla las herramientas empleadas al momento de crear la aplicación web y proporciona un análisis exhaustivo de su respectiva selección.

### 6.1.1.1 Stack de tecnologías

**6.1.1.1.1 Angular.** Al momento de realizar la elección de tecnologías para el front-end del aplicativo, Angular es considerado una de las mejores opciones debido a sus enormes ventajas y excelentes características. Su arquitectura modular ofrece escalabilidad a lo largo del tiempo y una estructura más ordenada y mantenible. Angular también cuenta con amplio soporte y con actualizaciones periódicas.

**6.1.1.1.2 Bootstrap.** Bootstrap es un framework CSS open source utilizado para el desarrollo front-end. Este framework ofrece herramientas y componentes predefinidos que permiten la creación de aplicaciones web responsive y con un diseño atractivo sin tanta complejidad. Trabajar con Bootstrap significa un ahorro de tiempo, ya que no hay que iniciar desde cero en el diseño y desarrollo de las vistas. Además, se esperaba tener un diseño consistente en todas las secciones del aplicativo.

**6.1.1.1.3 PostgreSQL y Sequelize.** PostgreSQL es un sistema de bases de datos de código abierto, altamente estable, que proporciona soporte a diferentes funciones de SQL (kinsta, 2023). PostgreSQL posee data types (tipos de datos) avanzados y permite ejecutar optimizaciones de rendimiento avanzadas, que son características que por lo general solo se ven en sistemas de bases de datos comerciales (Santiago, B., 2019). Para el proyecto, se utilizó un ORM para mapear los objetos a tablas en una estructura relacional. Sequelize es un ORM para NodeJS que nos permite manipular varias bases de datos SQL de una manera bastante sencilla, nos permite convertir tablas de una base de datos en entidades en un lenguaje de programación orientado a objetos (Alejandro. G., 2018).

**6.1.1.1.4 NodeJS y Express Node.js.** Es un entorno controlado por eventos diseñado para crear aplicaciones escalables, que permite establecer y gestionar múltiples conexiones al mismo tiempo (Chiyana, S., 2021). Express proporciona un conjunto de herramientas para aplicaciones web, peticiones y respuestas HTTP, enrutamiento y middleware para construir y desplegar aplicaciones a gran escala y preparadas para la empresa (kinsta, 2022). El uso de estas 2 tecnologías en conjunto ofrece al aplicativo eficiencia, rendimiento, un ecosistema sólido y la capacidad de usar JavaScript en ambos lados. Estas ventajas hacen de esta combinación una sobresaliente opción para trabajar la aplicación en la parte de back-end.

#### **6.1.1.2 Control de tareas**

**6.1.1.2.1 Jira.** Jira es una aplicación web que con el tiempo se ha convertido en el estándar del mercado en las áreas de gestión de proyectos, gestión de tareas y gestión de errores (Ionos, 2022). Jira es una herramienta poderosa para el control de tareas en el desarrollo colaborativo de software, ya que proporciona tableros de seguimiento, asignación y priorización de tareas. Hacer uso de este tipo de herramienta permite mantener un conocimiento y control eficiente y estructurado sobre el desarrollo de la aplicación. En este caso, se utilizó la versión gratuita del software.

#### **6.1.1.3 Versionamiento**

**6.1.1.3.1 Git y Github.** Git es un sistema de control de versiones distribuido gratuito y de código abierto diseñado (Git, s.f.). De este modo es posible tener un registro de todos los cambios en el proyecto y regresar a una versión anterior en caso de ser necesario. GitHub es una plataforma basada en la web donde los usuarios pueden alojar repositorios Git (Ezequiel, C., 2021).

**6.1.1.4 Testing.** El objetivo principal de realizar pruebas a la aplicación es garantizar la calidad del software y asegurar que cumple con todos los requerimientos establecidos. Para validar el correcto funcionamiento, se ejecutaron pruebas en diferentes casos de uso, como pruebas de integración, unitarias y pruebas de aceptación. Una vez finalizadas las pruebas, se realizó un análisis a los resultados con el fin de evaluar el rendimiento del aplicativo y solucionar los errores encontrados antes de su implementación.

- **Pruebas de integración:** las pruebas de integración están a cargo del examen de las interfaces entre los subsistemas o los grupos de componentes del programa o aplicación que se analiza, lo que contribuye a garantizar su funcionamiento correcto (KeepCoding, 2023).
- **Pruebas unitarias:** Las pruebas unitarias consisten en aislar una parte del código y comprobar que funciona a la perfección (YeePLY, s. f.). Estas pruebas son ejecutadas por el equipo de desarrollo y prueba las funcionalidades más pequeñas del software.
- **Pruebas de aceptación:** Las pruebas de aceptación consisten en descubrir que el software que ha creado no es lo que el cliente y los usuarios esperaban (nimble, s. f.). El objetivo de estas pruebas es confirmar que el producto desarrollado cumpla con los requerimientos.

#### ***6.1.1.4.1 MochaJS.***

Mocha es un framework de pruebas de JavaScript que se ejecuta en Node.js. Nos da la posibilidad de crear tanto tests síncronos como asíncronos de una forma muy sencilla (Rafael, M.,2017). Mocha ofrece flexibilidad ya que acepta distintos estilos de escritura para las pruebas y tienen una gran oferta de plugins y funciones.

**6.1.1.4.2 ChaiJS.** Es una librería de aserciones, la cual se puede emparejar con cualquier marco de pruebas de Javascript (Rafael, M.,2017). Chai se utiliza juntamente con Mocha buscando un entorno de pruebas más completo para la aplicación. Permite elegir el estilo que mejor se adapte a las preferencias y escritura de las pruebas.

**6.1.1.5 Despliegue.** Para esta etapa del proyecto se realizaron consideraciones sobre diferentes alternativas de alojamiento gratuito para el despliegue de una aplicación basada en el conjunto de tecnologías establecidas (PostgreSQL, Express, Angular, NodeJS). El objetivo era permitir el acceso remoto a la aplicación y permitir que usuarios desde cualquier lugar pudieran acceder sin ningún problema.

**6.1.1.5.1 Render.** Render es una plataforma de servicios en la nube que permite a los desarrolladores y los equipos de desarrollo alojar y exponer sin esfuerzo aplicaciones web y servicios en la nube, ya que ofrece toda la infraestructura esencial. Esta plataforma ofrece la posibilidad de integraciones con GitHub y GitLab y permite escalar automáticamente los recursos para no afectar el rendimiento de la aplicación.

## **6.1.2 Análisis previo**

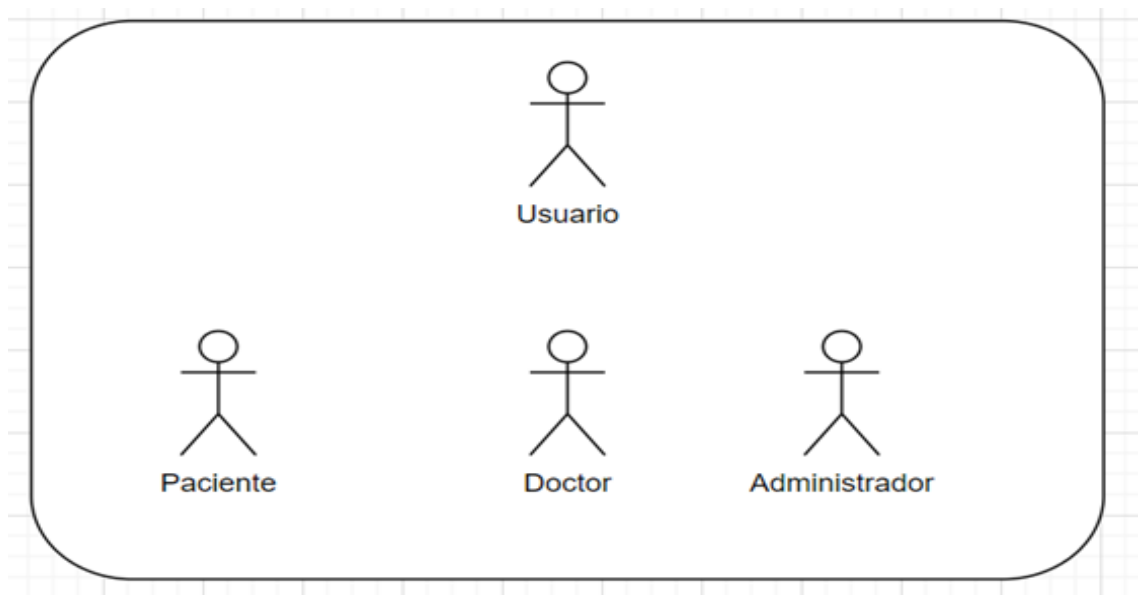
Previo a la fase de desarrollo, se realizó un análisis de los requerimientos, actores, casos de uso, actividades y módulos que hacen parte de la aplicación.



**6.1.2.1. Análisis de requerimientos.** Se darán a conocer los requerimientos funcionales y no funcionales de la aplicación, con el fin de mostrar las necesidades, funcionalidades y restricciones que se planearon para cumplir con este sistema y poder cumplir las expectativas.

**Figura 5.**

*Actores del Sistema*



### 6.1.2.2 Requerimientos Funcionales

**Tabla 1.**

*Requerimiento funcional 1*

<b>Requerimiento 1: Crear Cuenta</b>
<b>Descripción:</b> Los usuarios se registran en el sistema y escogen el rol ya sea paciente o doctor, para poder acceder a los servicios.
<b>Responsable:</b> Paciente y Doctor
<b>Entradas:</b> Datos personales del usuario como: Nombre, apellidos, documento entre otros

---

**Resultado:** Se agrega un registro a la base de datos y el usuario podrá acceder a todos los servicios

---

**Tabla 2.**

*Requerimiento funcional 2*

---

**Requerimiento 2: Información Previa antes del examen**

---

**Descripción:** Los usuarios pueden observar la información previa de los exámenes como, por ejemplo: preparación, cuidados entre otras.

---

**Responsable:** Paciente

---

**Entradas:** Datos del examen

---

**Resultado:** El usuario tendrá información previa antes de cada examen

---

**Tabla 3.**

*Requerimiento funcional 3*

---

**Requerimiento 3: Descargar Resultado**

---

**Descripción:** Los usuarios que ya están registrados pueden descargar los resultados de sus exámenes

---

**Responsable:** Paciente y Doctores

---

**Entradas:** Datos del examen como: Fecha, nombre y PDF con el resultado

---

**Resultado:** Los usuarios podrá observar de manera clara los resultados y podrán dar un diagnóstico

---

**Tabla 4.**

*Requerimiento funcional 4*

---

**Requerimiento 4: Cargar Resultado**

---

---

**Descripción:** El administrador sube los resultados al sistema, esta acción también la puede realizar el doctor

---

**Responsable:** Administrador y Doctor

---

**Entradas:** Datos del examen como: Fecha, nombre y PDF con el resultado

---

**Resultado:** Resultados cargados y disponibles para la visión de los usuarios

---

*Tabla 5.*

*Requerimiento funcional 5*

---

**Requerimiento 5: Recuperar contraseña**

---

**Descripción:** El usuario puede recuperar su contraseña siguiendo el debido proceso para su restauración

---

**Responsable:** Doctor y pacientes

---

**Entradas:**

---

**Resultado:** Token para restaurar la nueva contraseña

---

*Tabla 6.*

*Requerimiento funcional 6*

---

**Requerimiento 6: Actualizar contraseña**

---

**Descripción:** El usuario cambia su contraseña cuando lo vea necesario sin embargo debe cumplir con los requisitos establecidos.

---

**Responsable:** Doctor y pacientes

---

**Entradas:** Contraseña que desea cambiar

---

**Resultado:** Contraseña nueva

---

### 6.1.2.3 Requerimientos no Funcionales:

**Tabla 7.**

*Requerimiento no funcional 1*

<b>Requerimiento 7: Protocolos de seguridad</b>
<b>Descripción:</b> El sistema debe contar con un conjunto de protocolos de seguridad que permita al usuario confiar en el software y así lograr el mayor nivel de integridad de la información.
<b>Responsable:</b> Equipo de desarrollo.
<b>Entradas:</b>
<b>Resultado:</b> Se confiabilidad de la información.

**Tabla 8.**

*Requerimiento no funcional 2*

<b>Requerimiento 8: Interfaz de colores</b>
<b>Descripción:</b> El sitio web debe poseer interfaces gráficas bien formadas y llevar en su interfaz los colores adecuados para un laboratorio clínico
<b>Responsable:</b> Equipo de desarrollo.
<b>Entradas:</b>
<b>Resultado:</b> Interfaz con los colores representativos del equipo.

**Tabla 9.**

*Requerimiento no funcional 3*

<b>Requerimiento 9: Disponibilidad tiempo completo</b>
<b>Descripción:</b> El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente acceder.
<b>Responsable:</b> Equipo de desarrollo

---

**Entradas:**

---

**Resultado:** El sistema se encuentra en la web todo el tiempo.

---

**Tabla 10.**

*Requerimiento no funcional 4*

---

**Requerimiento 10: Soporte de usuarios**

---

**Descripción:** El sistema debe ser capaz de operar adecuadamente con hasta 100.000 usuarios con sesiones concurrentes.

---

**Responsable:** Sistema.

---

**Entradas:**

---

**Resultado:** El sistema soporta 10000 usuarios conectados simultáneamente

---

**Tabla 11.**

*Requerimiento no funcional 5*

---

**Requerimiento 11: tiempo de respuesta**

---

**Descripción:** Toda funcionalidad del sistema y transacción de negocio debe responder al usuario en menos de 5 segundos.

---

**Responsable:** Sistema.

---

**Entradas:**

---

**Resultado:** La página tiene un tiempo de respuesta inferior a los 5 segundos

---

**Tabla 12.**

*Requerimiento no funcional 6*

---

**Requerimiento 12: Control de errores**

---

**Descripción:** El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final

---

---

**Responsable:** Sistema, Equipo de desarrollo

---

**Entradas:**

---

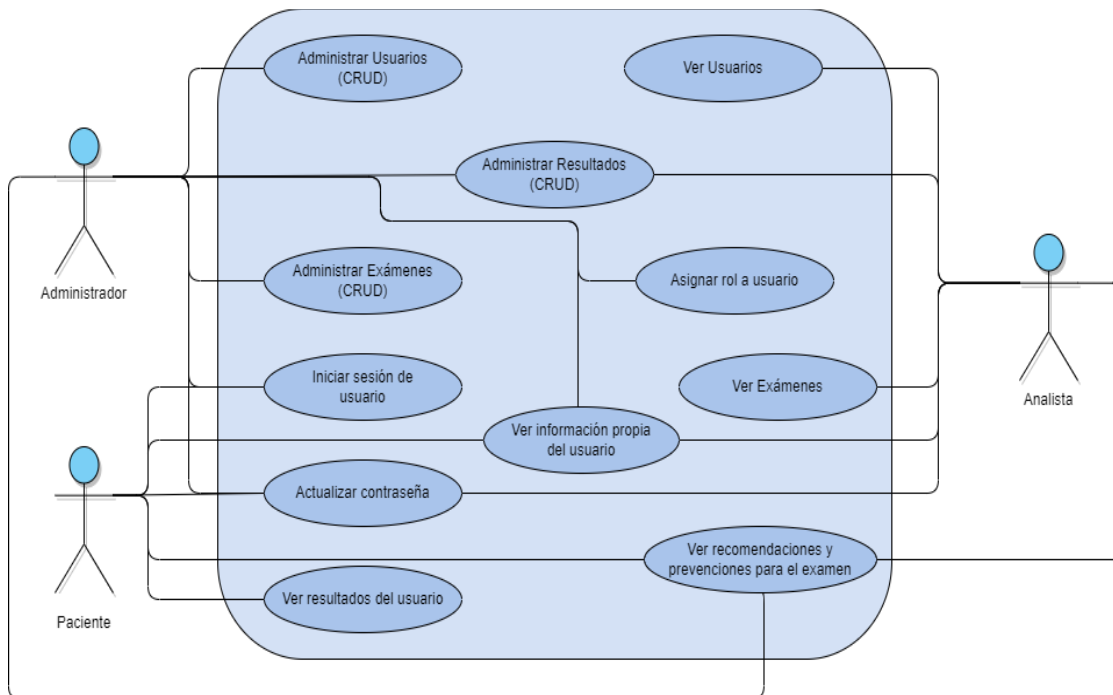
**Resultado:** Control de errores en un tiempo menor a 4 horas

---

#### 6.1.2.4 Casos de Uso.

**Figura 6.**

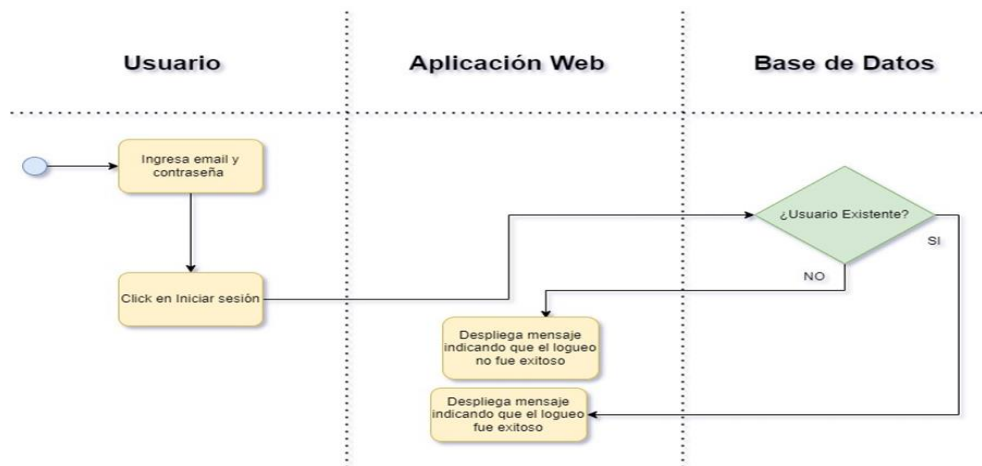
*Diagrama de casos de uso*



**6.1.2.5 Diagramas de Actividades.** Se mostrarán los diagramas de actividades que representan las principales acciones o tareas que puede realizar el sistema. Estos diagramas visualizan de manera gráfica cómo fluye el proceso de cada funcionalidad y que acciones se llevan a cabo en cada paso.

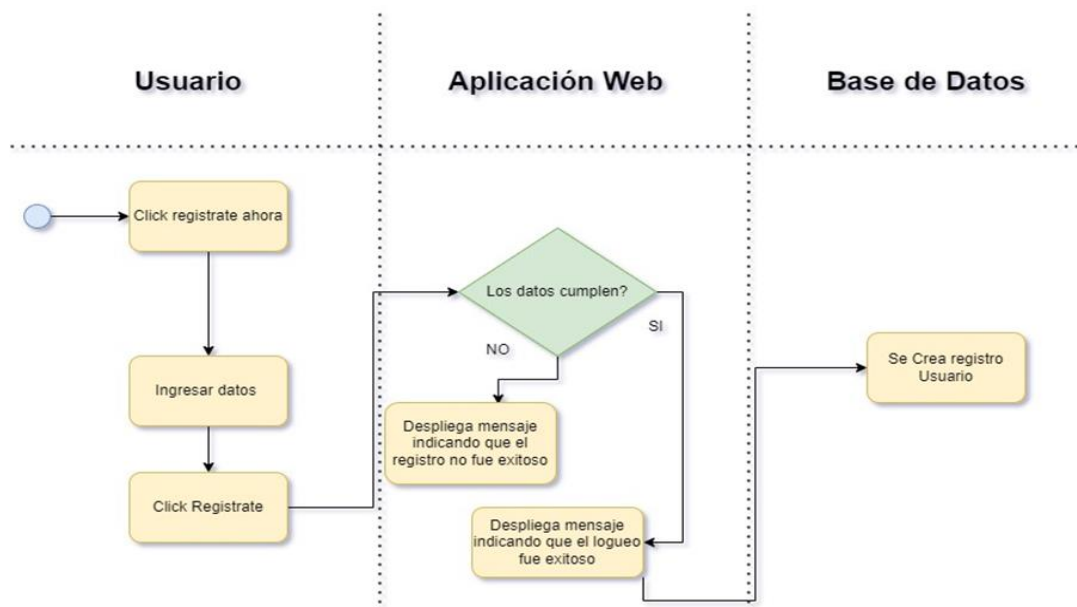
**Figura 7.**

*Proceso para iniciar sesión, es el mismo para cada rol de usuario*



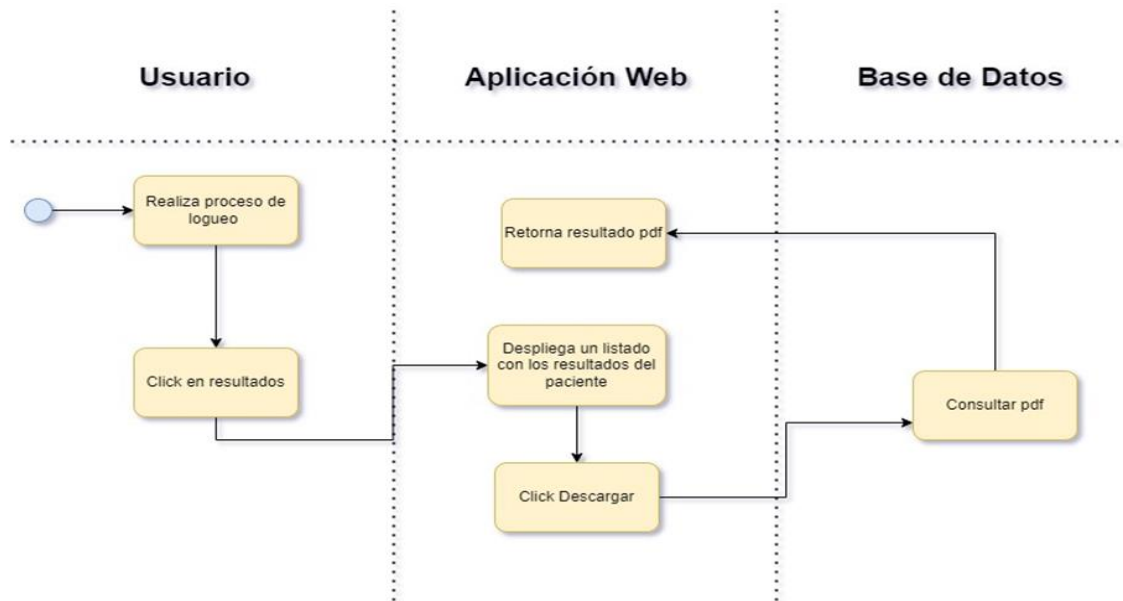
**Figura 8.**

*Registro creación de cuenta*



**Figura 9.**

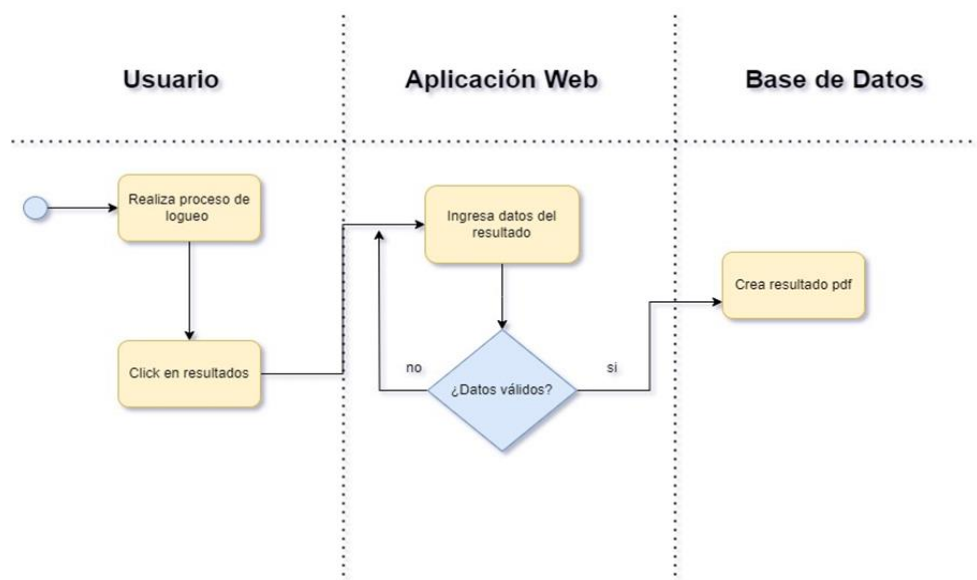
*Ver resultados*





**Figura 10.**

*Subir resultados*



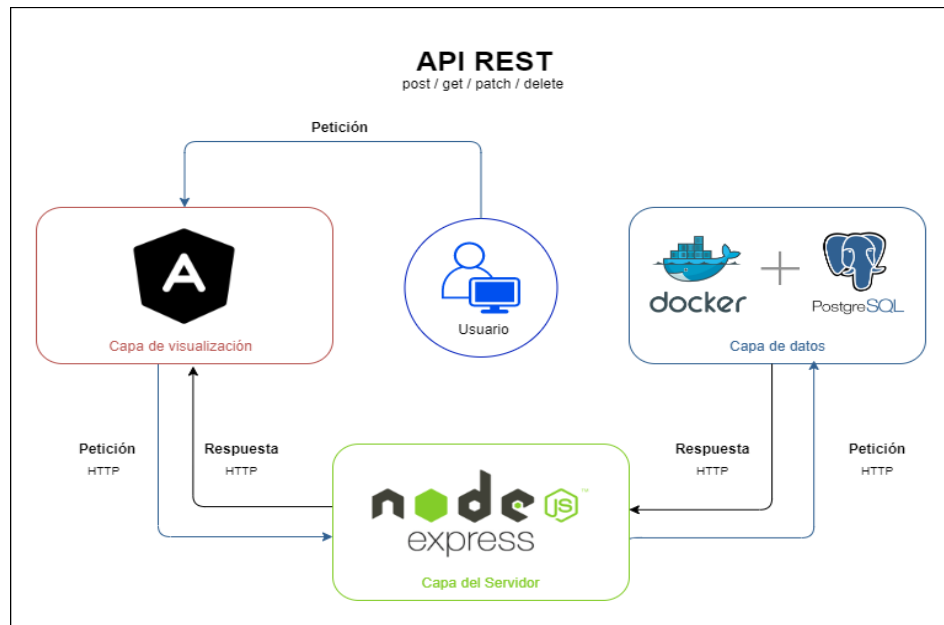
**6.1.2.6 Arquitectura.** La arquitectura del proyecto se basa en una arquitectura de tres capas que permite un funcionamiento eficiente y escalable. En primer lugar, contamos con la capa de datos, un componente crucial del proyecto, ya que garantiza la persistencia y disponibilidad de los datos. La capa del servidor es esencial para implementar las funcionalidades del sistema. Toda la lógica de negocio y el procesamiento de datos residen aquí, gestiona las solicitudes y consultas recibidas desde la capa de visualización. La interfaz de usuario está construida en la capa de visualización, que es responsable de presentar la información al usuario final. Las respuestas necesarias para mostrar al usuario se reciben aquí a través de peticiones HTTP, que se envían al servidor. En esta capa se implementan las interacciones necesarias para una experiencia fluida y amigable.

El uso de una arquitectura de tres capas permite desarrollar un sistema confiable y escalable. Este enfoque proporciona una solución estructurada y modular que garantiza la separación de responsabilidades, facilitando así el mantenimiento y desarrollo del proyecto en el

tiempo. El principal beneficio de la arquitectura de tres niveles es su separación lógica y física de la funcionalidad (IBM, s. f.).

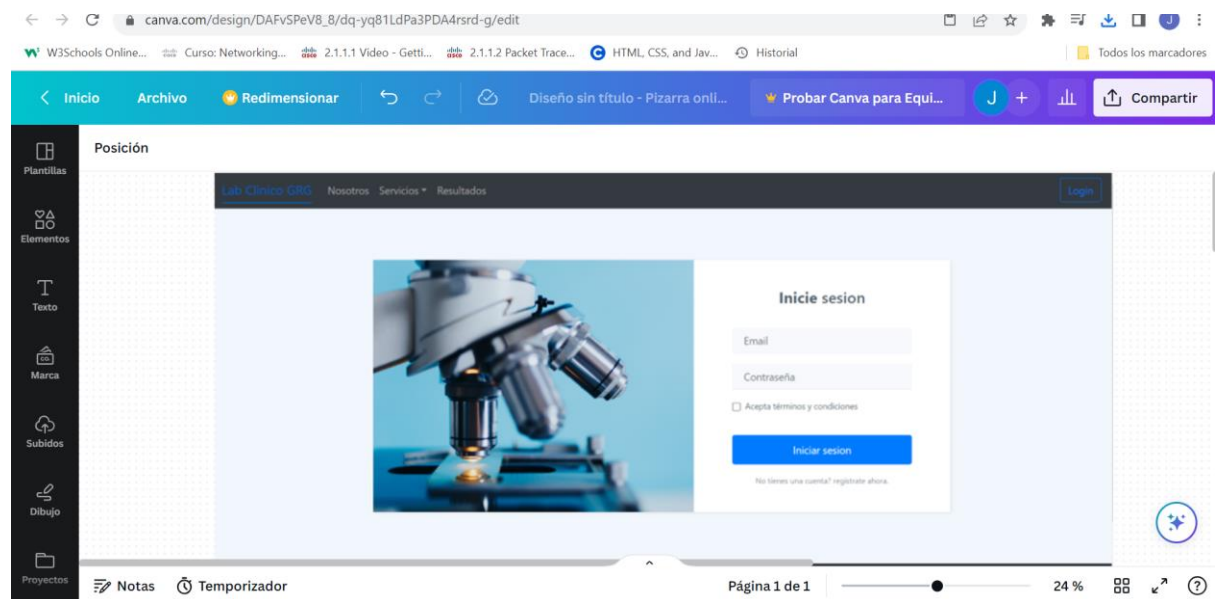
**Figura 11.**

*Arquitectura*



### 6.1.3 Diseño Interfaz de Usuario

Se mostrará el proceso que se realizó para el diseño de las ventanas del aplicativo, se utilizó el aplicativo Canva que sirve para realizar estos prototipos, se buscó con el propósito de que la página fuera bastante estética e intuitiva para los usuarios que van a navegar en ella.

**Figura 12.****Mockup de interfaz para iniciar sesión****6.2 Fase de desarrollo**

La fase de desarrollo se dividió en 3 etapas, 2 para el desarrollo del aplicativo y 1 para el curso. La fase del aplicativo consta del desarrollo Front-end y Back-end.

**6.2.1 Avance Sprints**

A continuación, se puede observar la configuración y desarrollo del proyecto a partir de cada sprint desarrollado.

**Sprint 1**

Para el inicio del proyecto hubo contactos con personas relacionadas con el área de negocio de los exámenes de laboratorio clínico, esto con el fin de conocer las principales necesidades que debe cumplir un aplicativo para un laboratorio clínico. Con esto se esperaba tener más conocimiento del área y así:

- Definir requerimientos del usuario.
- Definir tecnologías para el desarrollo de la aplicación.
- Crear diagramas para casos de uso y actividades.
- Realizar mockups para las vistas de la aplicación.

**Figura 13.**

### *Tareas sprint 1*

▼

Tablero Sprint 1

13 feb – 26 feb (8 incidencias)

0

0

0

Completar sprint

...

- Definir requerimientos del usuario. - Definir tecnologías para el desarrollo de la aplicación. - Crear diagramas para casos de uso y actividades. - Realizar mockups para las vistas de la aplicación.

tAB-1

Crear diagrama de actividades

FINALIZADA ▼

FJ

tAB-2

Crear diagrama de casos de uso

FINALIZADA ▼

J

tAB-3

Diseñar mockup para la vista 'home'

FINALIZADA ▼

FJ

tAB-4

Diseñar mockup para la vista 'exámenes'

FINALIZADA ▼

FJ

tAB-5

Diseñar mockup para la vista 'login'

FINALIZADA ▼

FJ

tAB-6

Diseñar mockup para la vista 'registro de usuario'

FINALIZADA ▼

J

tAB-7

Diseñar mockup para la vista 'resultados'

FINALIZADA ▼

J

tAB-8

Definir tecnologías para el backend, frontend y almacenamiento de datos

FINALIZADA ▼

J

## **Sprint 2**

Una vez completadas las historias del sprint 1, con base en los mockups diseñados, se procede con el desarrollo de todas las vistas planteadas para la aplicación con el framework establecido.

**Figura 14.**

### *Tareas sprint 2*

▼ Tablero Sprint 2 27 feb – 12 mar (5 incidencias)

000

Completar sprint

⋮

- Desarrollar todas las vistas para la aplicación con el framework establecido (Angular);

■ tAB-9

Desarrollar vista 'home'

FINALIZADA ▼

FJ

■ tAB-10

Desarrollar vista 'login'

FINALIZADA ▼

FJ

■ tAB-11

Desarrollar vista 'registro de usuario'

FINALIZADA ▼

FJ

■ tAB-12

Desarrollar vistas 'resultados'

FINALIZADA ▼

J

■ tAB-13

Desarrollar vista 'exámenes'

FINALIZADA ▼

J

### Sprint 3

Para el sprint 3, se establece el objetivo de crear y montar toda la estructura del proyecto en Angular. Se definen los módulos y componentes a crear con el respectivo routing.

**Figura 15.**

*Tareas sprint 3*

▼ Tablero Sprint 3 13 mar – 26 mar (3 incidencias)		0 0 0	Completar sprint	...
- Creación de estructura en Angular				
LAB-16	Creación de rutas hijas y LazyLoading	FINALIZADA ▼	J	
LAB-15	Creación de modulos en Angular	FINALIZADA ▼	J	
LAB-14	Creación de componentes en Angular	FINALIZADA ▼	FJ	

### Sprint 4

Con la estructura front-end y las vistas completas, se da inicio a la configuración inicial del entorno de desarrollo back-end. Además, se procede a implementar los routers correspondientes a cada módulo.

**Figura 16.**

*Tareas sprint 4*

▼ Tablero Sprint 4 27 mar – 9 abr (4 incidencias)		0 0 0	Completar sprint	...
- Configuración del entorno de desarrollo backend. - Configuración de router para los módulos.				
LAB-17	instalación y configuración de librerías, dependencias, y herramientas	FINALIZADA ▼		
LAB-18	Routing para usuarios con express.js	FINALIZADA ▼		
LAB-19	Routing para exámenes con express.js	FINALIZADA ▼		
LAB-20	Routing para resultados con express.js	FINALIZADA ▼		

### Sprint 5

Para el sprint 5, se planteó el objetivo de diseñar e implementar una estructura que permita el correcto control de errores. Con el manejo de errores finalizado, inició la creación de los primeros servicios.

- Control y manejo de errores.
- Creación de middlewares.
- Creación de servicios.

**Figura 17.**

*Tareas sprint 5*

▼ Tablero Sprint 5 10 abr – 23 abr (4 incidencias)		0 0 0	Completar sprint	...
- Control y manejo de errores. - Creación de middlewares. - Creación de servicios.				
LAB-21	Manejo de errores con middlewares	FINALIZADA	▼	👤
LAB-22	Manejos de errores con boom	FINALIZADA	▼	👤
LAB-23	Validación de datos con joi	FINALIZADA	▼	👤
LAB-24	Crear servicios para resultados y exámenes	FINALIZADA	▼	👤

**Sprint 6**

En este sprint, el equipo continuó con toda la parte de los datos de la aplicación, así que se consiguieron los siguientes avances:

- Persistencia de datos en node.js.
- Configuración de postgres, docker y sequelize

**Figura 18.**

*Tareas sprint 6*

Clave	Resumen	Tipo de incidencia	Epic	Estado	Persona asignada
LAB-25	instalación de docker	Historia		FINALIZADA	J
LAB-26	Configuracion de postgres en docker	Historia		FINALIZADA	J
LAB-27	Configuracion de pgadmin	Historia		FINALIZADA	FJ
LAB-28	Integración node - postgres	Historia		FINALIZADA	J
LAB-29	Instalacion y configuracion de sequelize	Historia		FINALIZADA	FJ
LAB-30	CRUD con sequelize	Historia		FINALIZADA	FJ

## Sprint 7

En este sprint, el equipo enfrentó varios desafíos clave relacionados con la creación de los modelos de datos con base en los requerimientos, migraciones con Sequelize y la definición de relaciones entre modelos. Estos desafíos se abordaron con éxito, lo que permitió que el proyecto logrará un progreso significativo.

**Figura 19.**

### Tareas sprint 7

▼

Tablero Sprint 7 8 may – 21 may (4 incidencias)

000

Completar sprint

...

- Creación de modelos de datos. - migraciones con sequelize - definir relaciones

TAB-31

Crear modelos de datos en sequelize

FINALIZADA ▼

FJ

TAB-32

Crear migraciones en sequelize

FINALIZADA ▼

FJ

TAB-33

Definir relaciones entre entidades

FINALIZADA ▼

FJ

TAB-34

filtrar resultados por condiciones

FINALIZADA ▼

J

## Sprint 8

Una vez se logró establecer una comunicación fluida con la base de datos, se crearon todos los servicios necesarios para implementar las funcionalidades con cada módulo.

**Figura 20.**

### Tareas sprint 8

▼ Tablero Sprint 8 22 may – 4 jun (3 incidencias)

0

0

0

Completar sprint

...

TAB-36

Creación de servicios crud para usuarios

- FINALIZADA ▼

TAB-38

Creación de servicios crud para exámenes

- FINALIZADA ▼

TAB-37

Creacion de servicios crud para resultados

- FINALIZADA ▼

## Sprint 9

En este sprint, se logró implementar con éxito las siguientes funcionalidades clave:

- Autenticación y autorización de usuarios, permitiendo a los usuarios iniciar sesión y verificar sus credenciales.
- Manejo y verificación de roles de usuarios, asegurando que cada usuario tenga los privilegios adecuados.
- Protección y seguridad de datos y rutas.

Se trabajó con JWT, passport.js y middlewares para cumplir todas las historias del sprint.

**Figura 21.**

### *Tareas sprint 9*

▼ Tablero Sprint 9 6 jun – 25 jun (7 incidencias)			0 0 0	Completar sprint	...
📌	🔖 tAB=39	Autenticación de usuarios	FINALIZADA ▼	👤 FJ	
📌	🔖 tAB=40	Crear middleware verificacion	FINALIZADA ▼	👤 FJ	
📌	🔖 tAB=41	Implementar login con passport	FINALIZADA ▼	👤 FJ	
📌	🔖 tAB=42	Firmar y verificar tokens	FINALIZADA ▼	👤 J	
📌	🔖 tAB=43	generar jwt en el servicio	FINALIZADA ▼	👤 J	
📌	🔖 tAB=44	protección de rutas	FINALIZADA ▼	👤 J	
📌	🔖 tAB=45	Control de roles	FINALIZADA ▼	👤 J	

## **Sprint 10**

El objetivo de este sprint era configurar la aplicación de envío automáticos de mensajes por correo electrónicos para la notificación de nuevos resultados clínicos disponibles y la recuperación de contraseña en caso de olvido. Así mismo, se inició el plan de contenidos para el curso.



**Figura 22.**

*Tareas sprint 10*

Clave	Resumen	Tipo de incidencia	Epic	Estado	Persona asignada
LAB-46	Implementar y configurar Nodemailer.js	Historia		FINALIZADA	FJ
LAB-47	Aplicar envío de correos automáticos según los requeri...	Historia		FINALIZADA	FJ
LAB-48	Validación de token para actualización de contraseña	Historia		FINALIZADA	FJ
LAB-49	Inicio desarrollo plan para el curso	Historia		FINALIZADA	J

## Sprint 11

Con base en el plan de trabajo diseñado para el curso se inició con la definición de contenidos y materiales para el curso, se diseñó un cronograma con base en las semanas del semestre. Se busca crear estrategias con el fin de incentivar la creatividad de los estudiantes.

**Figura 23.**

*Tareas sprint 11*

▼

Tablero Sprint 11

24 jul – 8 ago (6 incidencias)






0

0

0

Completar sprint

...

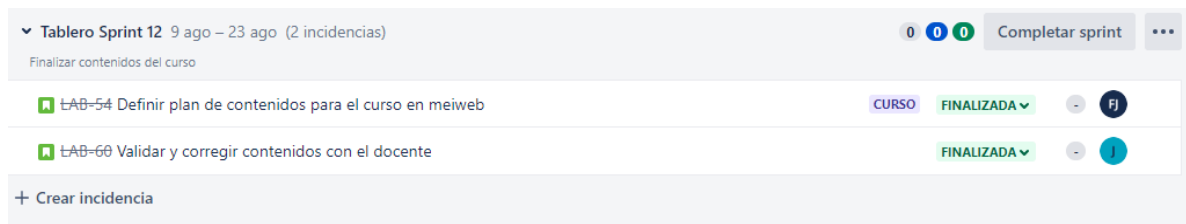
<input type="checkbox"/>	 LAB-54 Definir plan de contenidos para el curso en meiwab	CURSO	-	EN CURSO	
<input checked="" type="checkbox"/>	 LAB-55 Establecer los recursos y tecnologías a usar	CURSO	-	FINALIZADA	
<input type="checkbox"/>	 LAB-56 Iniciar el desarrollo del curso	CURSO	-	FINALIZADA	
<input type="checkbox"/>	 LAB-53 Definir Plan de cronograma para el curso en meiwab	CURSO	-	FINALIZADA	
<input type="checkbox"/>	 LAB-57 Análisis de implementación de actividades didácticas	CURSO	-	FINALIZADA	
<input type="checkbox"/>	 LAB-58 Definir como implementar e incentivar la creatividad en el proceso de aprendizaje	CURSO	-	FINALIZADA	

## Sprint 12

En este sprint se culmina con el plan de contenidos que se empezó a diseñar en el sprint anterior y en conversaciones con el docente se corrige y se validan el plan.

**Figura 24.**

*Tareas sprint 12*

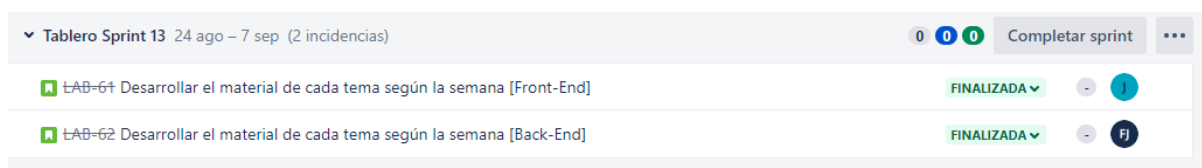


**Sprint 13**

En este sprint se realiza todo el material para el curso, con todo el contenido planteado inicialmente. Se organiza por semanas y se estructura en 2 partes, empezando por el front-end y finalizando con los temas del back-end. En el contenido de cada semana se encuentran recursos (enlaces a material de lectura, videos, blogs), un documento con el plan de trabajo y el material práctico para ejemplos y ejercicios propuestos.

**Figura 25.**

*Tareas sprint 13*

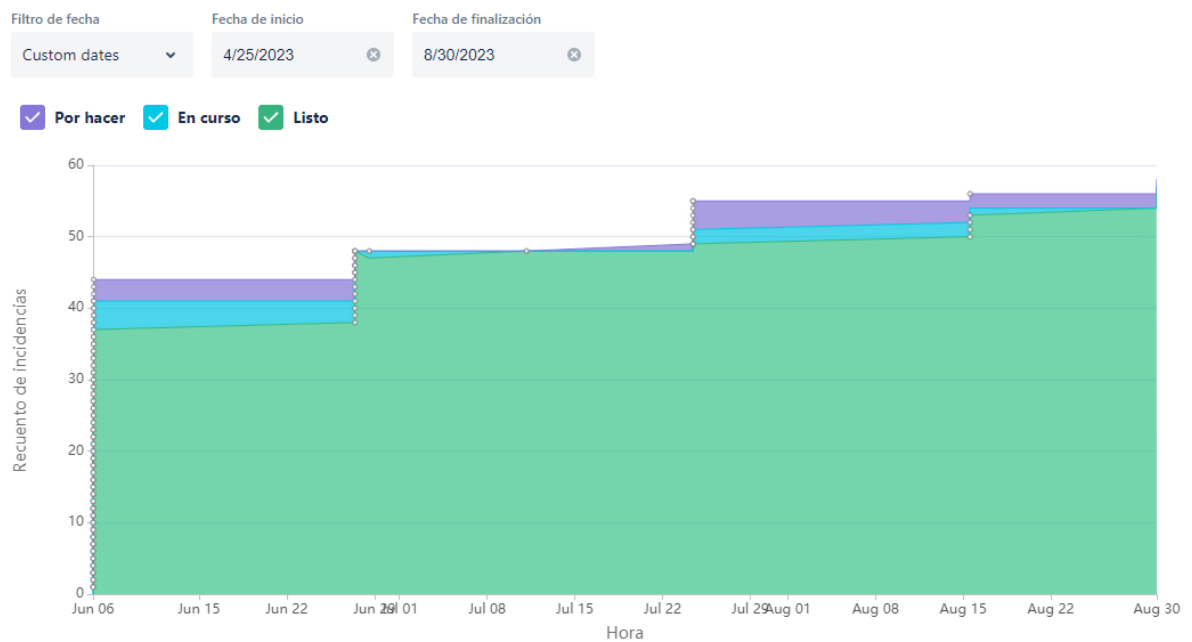


**Diagrama de flujo acumulado Jira**

En este diagrama se puede observar el avance del desarrollo a lo largo del proyecto en cuanto a las tareas propuestas. En este diagrama muestra una pequeña fracción, desde junio hasta finales de agosto, pero que sirve de referencia para entender el desarrollo del proyecto.

**Figura 26.**

*Diagrama de flujo acumulado*



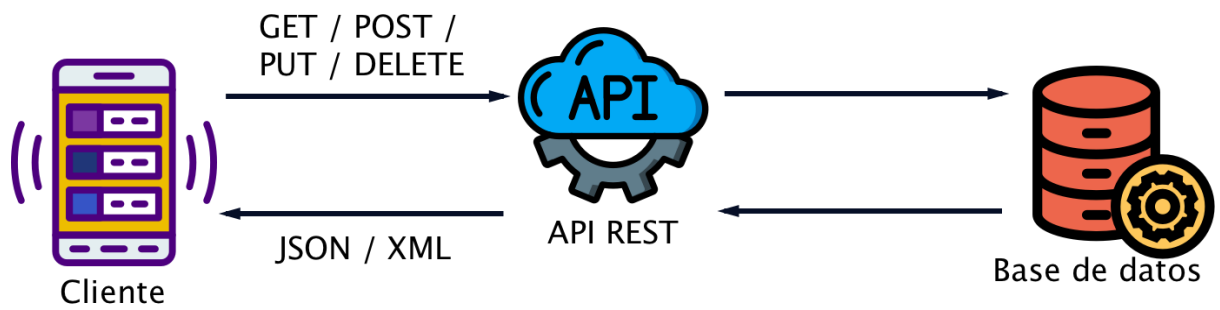
## 6.2.2 Aplicación Web Laboratorio Clínico

**6.2.2.1 Desarrollo Back-end.** En el back-end se trabajó con la arquitectura API Rest con el Stack de tecnologías, Node.js y express para la capa servidor. Para la capa de los datos, se utilizó PostgreSQL como base de datos y Sequelize como ORM (mapeo relacional de objetos).

Adicionalmente, se hizo uso de librerías disponibles en NPM con el fin de un desarrollo más eficiente.

**Figura 27.**

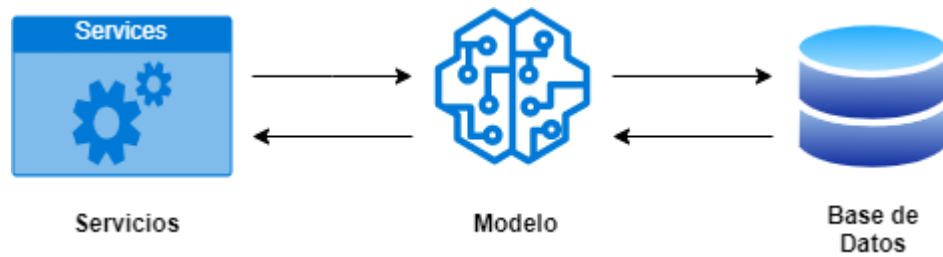
*Arquitectura API Rest*

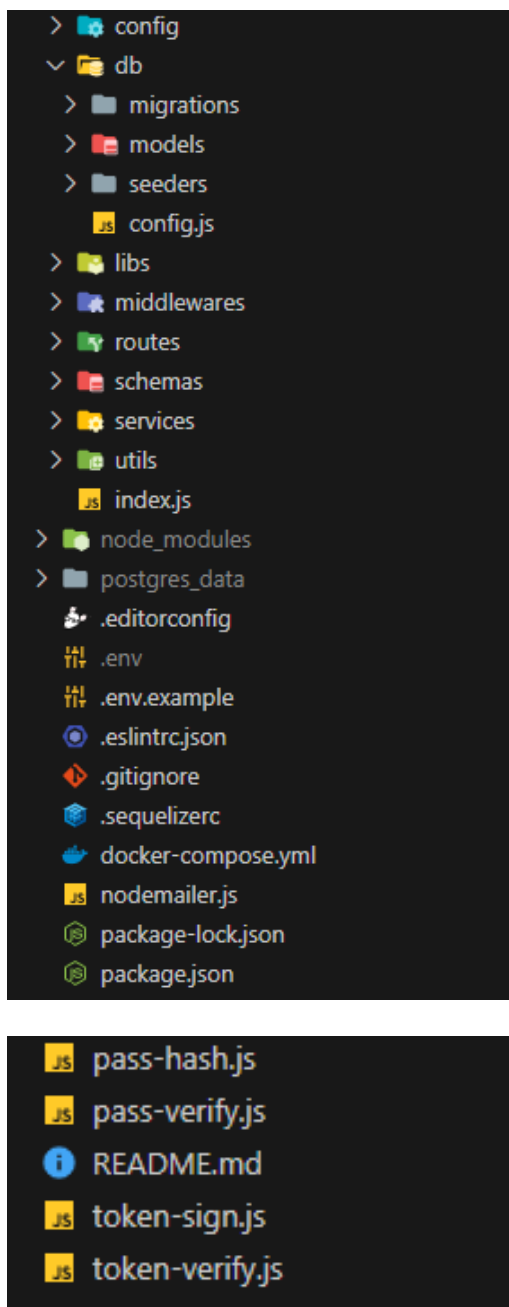


Adaptado de dossetenta.com

**Figura 28.**

*Estructura Back-end*



**Figura 29.***Estructura de Carpetas Back-end*

- **config:** Contiene el archivo que llama las variables de entorno.
- **db:** Contiene todo lo relacionado a la base de datos.
  - **migrations:** en esta carpeta se encuentra la historia de las migraciones a la base de datos.

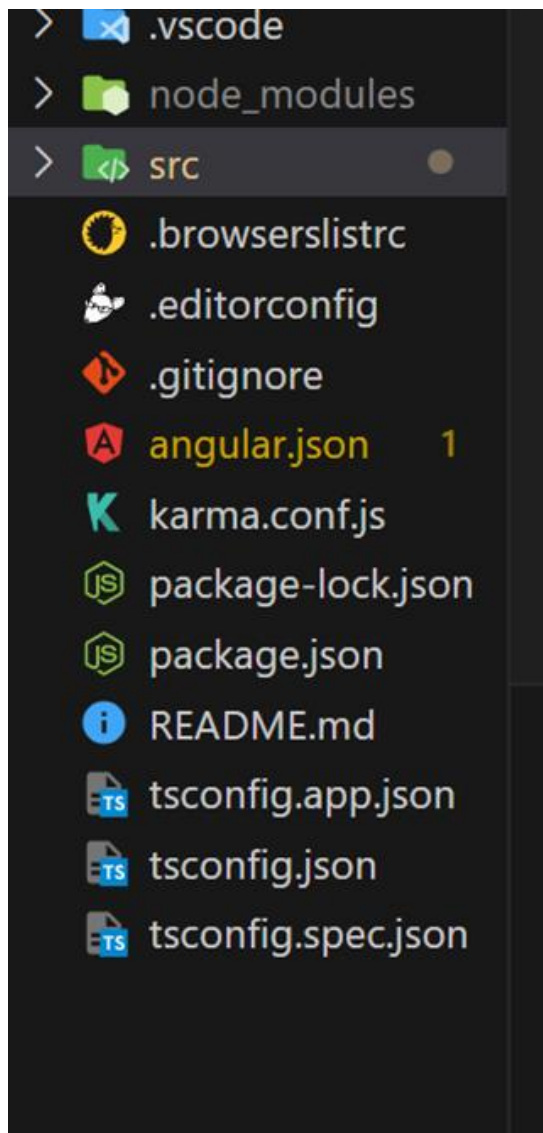
- **models:** se encuentran los modelos de datos que representan toda la lógica de la base datos.
- **seeders:** (vacía) se crea por defecto y son herramientas para poblar bases de datos.
- **libs:** conexión con la capa de datos.
- **middlewares:** middlewares de verificación, autenticación y control de errores.
- **routes:** se definen las rutas del aplicativo con express. Por cada entidad existe un router que representa las rutas.
- **schemas:** esquemas que validan que los datos ingresados cumplan con los requerimientos planteados.
- **services:** esta carpeta almacena los servicios encargados de manejar las peticiones http para cada endpoint.
- **utils:** dentro de la carpeta se encuentran las estrategias utilizadas para autorización y autenticación.
- **node\_modules:** Almacena las librerías y módulos externos que utiliza el proyecto
- **postgres\_data:** metadata de postgres.
- **.editorconfig:** contiene el formato para el editor de texto.
- **.env:** información y configuración de las variables de entorno.
- **.env.example:** archivo guía sobre cómo enviar las variables de entorno.
- **.eslintrc.json:** archivo de configuración de la librería eslint.
- **.gitignore:** archivo que especifica a git los archivos y directorios a omitir.

- **.sequelizerc:** archivo de configuración de sequelize.
- **docker-compose.yml:** contiene la configuración para ejecutar contenedores en docker.
- **nodemailer:** archivo de configuración para la dependencia nodemailer.
- **package-lock.json:** almacena las versiones de las dependencias instaladas.
- **package.json:** configuración principal del proyecto, contiene información detallada de todas las dependencias instaladas.
- **pass-hash & pass-verify:** responsables del proceso de hash y verificación de contraseñas.
- **README.md:** archivo que contiene información y documentación acerca del proyecto.
- **token-sign & token-verify:** archivos encargados de la generación y verificación de tokens.

### 6.2.2.1 Desarrollo Front-end.

*Figura 30.*

*Estructura Carpetas Inicial Front-end*



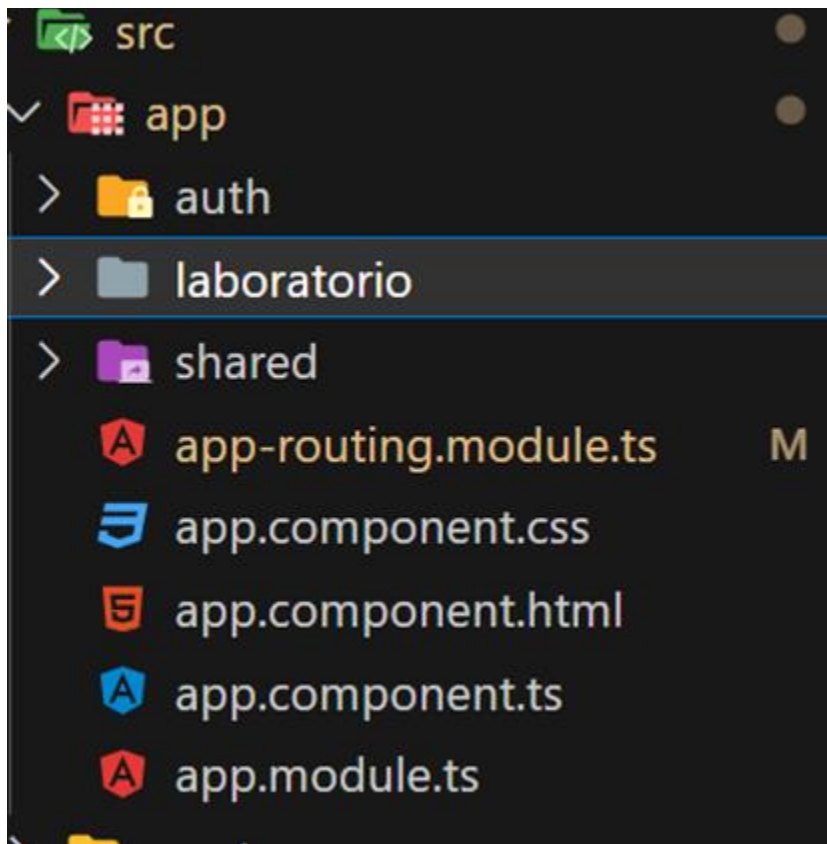
- **Browserslistrc:** Se utiliza para especificar la compatibilidad del navegador para la aplicación, en este archivo podemos definir una lista de navegadores y versiones para que el proyecto sea compatible.
- **Editorconfig:** Este archivo permite mantener y establecer consistencia del estilo en el que se va a codificar el proyecto.



- **.gitignore:** Se encarga de especificar los archivos o directorios que deben ser ignorados al subirlos a git.
- **angular.json:** Contiene información sobre la estructura del proyecto, las dependencias, las opciones de compilación y muchas otras configuraciones relacionadas con el desarrollo y la compilación del proyecto.
- **Karma.conf.js:** Sirve para configurar y personalizar el entorno de ejecución para las pruebas unitarias que se van a realizar
- **Package.json:** Se encarga de administrar dependencias, scripts, metadatos y otras configuraciones del proyecto.
- **Tsconfig.json:** es un archivo de configuración utilizado en proyectos de TypeScript para especificar las opciones y ajustes de compilación del compilador de TypeScript (tsc).

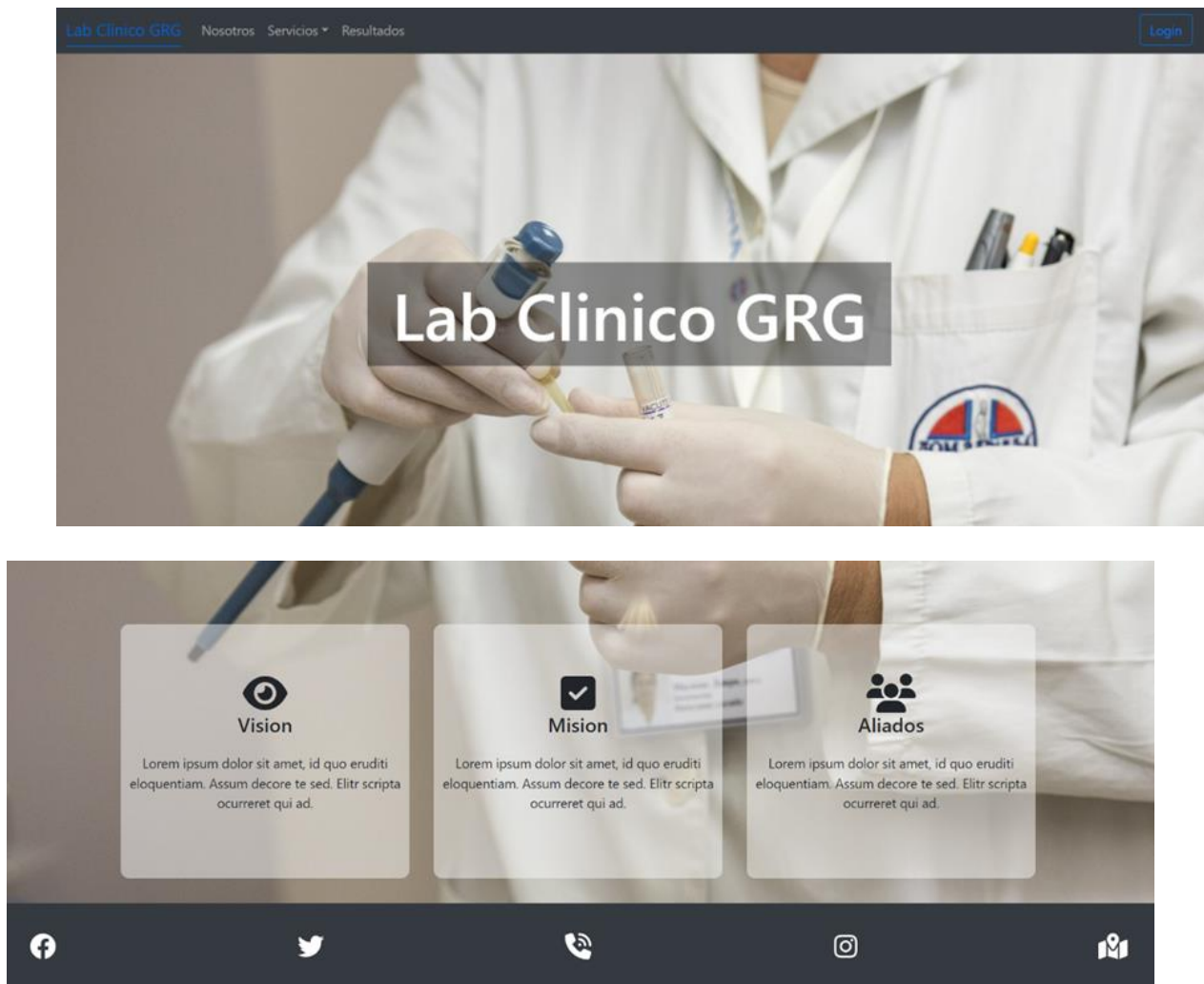
**Figura 31.**

*Estructura Front-end laboratorio*



En la carpeta src se encuentran los archivos de código fuente de la aplicación como los módulos, componentes, servicios, HTML, Css y de más archivos que integran el aplicativo.

- Esta es la página principal donde se podrá ver información básica de la empresa como su misión, visión y aliados, también podrá navegar por las diferentes secciones y visitar las redes sociales.

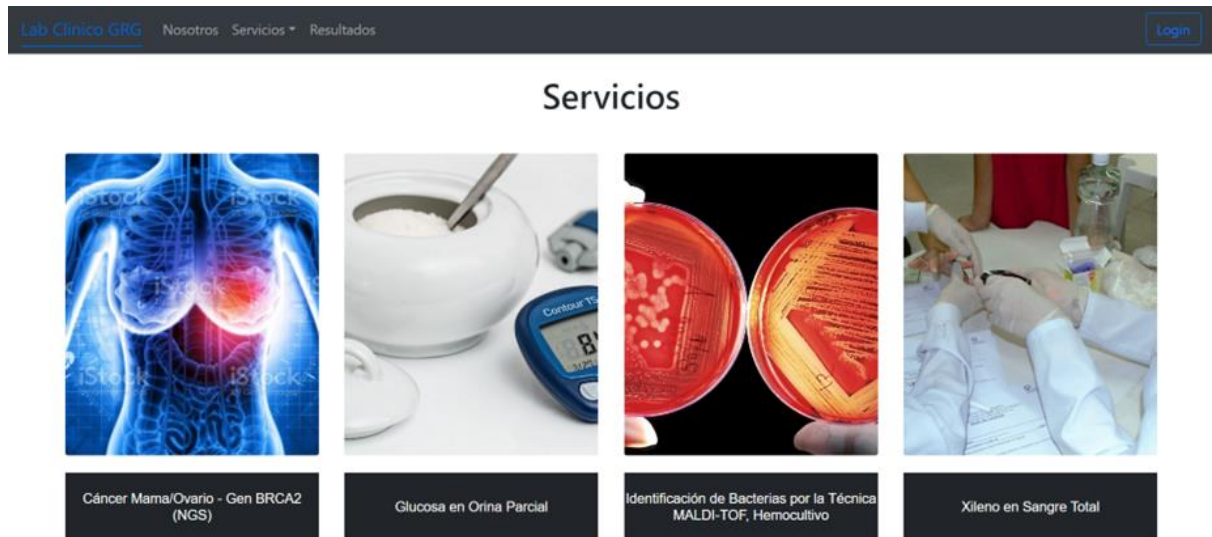
**Figura 32.***Vista Inicial home*

Cualquier persona podrá ver esta información no necesita estar logueado.

- En la sección de servicio el paciente podrá revisar información sobre los exámenes que presta el laboratorio clínico y breve información sobre él.

**Figura 33.**

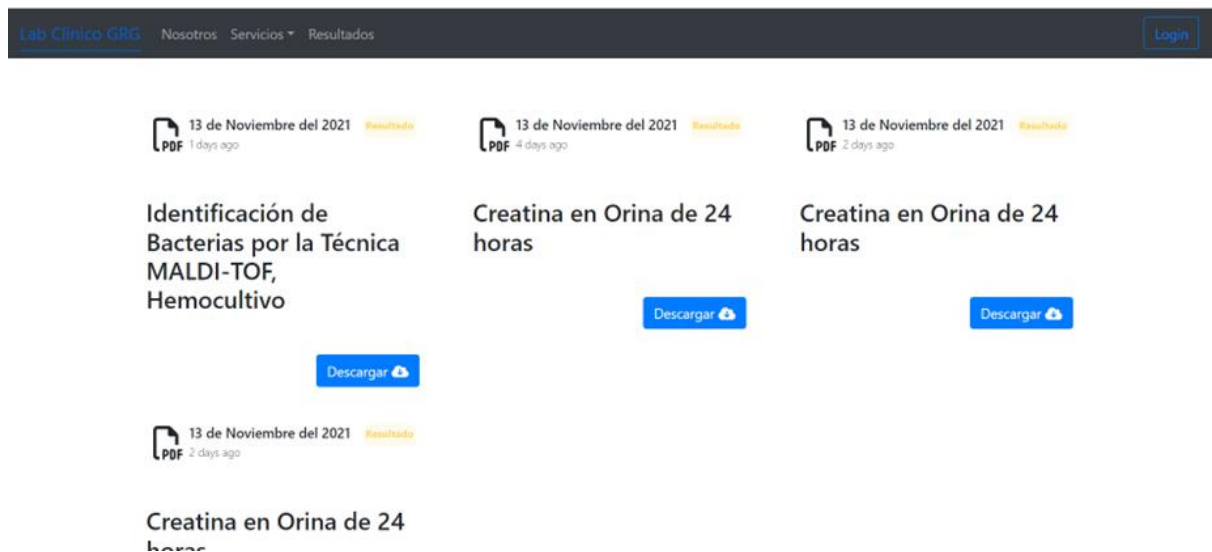
*Vista de servicios*



- En esta sección se podrán ver los resultados el cliente debe tener iniciada la sesión para acceder a esta ruta

**Figura 34.**

*Vista de resultados por paciente*



- Inicio de sesión

**Figura 35.**

*Vista inicio de sesión*

Lab Clínico GRG    Nosotros    Servicios    Resultados    Login

### Inicie sesion

Email

Contraseña

☐ Acepta términos y condiciones

Iniciar sesion

No tienes una cuenta? regístrate ahora.

- Registro

**Figura 36.**

*Vista registro de usuario*

### Crea una cuenta

Documento de identidad

Nombre completo

dd/mm/aaaa

Email

Contraseña

Confirmar contraseña

☐ Acepto los términos y condiciones.

Regístrate

Ya tienes una cuenta? Inicie sesion.

### 6.2.3 Curso Programación Web

En esta sección del documento, se explica el proceso de desarrollo del curso de programación web utilizando todo el material práctico se desarrolló con el aplicativo web para un laboratorio clínico.

**Figura 37.**

*Lista de tareas para el curso de programación web*

Incidencias completadas						<a href="#">Ver en el navegador de incidencias</a>	
Clave :	Resumen :	Tipo de incidencia :	Epic :	Estado :	Persona asignada :	Puntos de historia :	
<a href="#">LAB-53</a>	Definir Plan de cronograma para el curso en meiweb	 Historia	CURSO	FINALIZADA	FJ	-	
<a href="#">LAB-55</a>	Establecer los recursos y tecnologías a usar	 Historia	CURSO	FINALIZADA	FJ	-	
<a href="#">LAB-56</a>	Iniciar el desarrollo del curso	 Historia	CURSO	FINALIZADA	FJ	-	
<a href="#">LAB-57</a>	Análisis de implementación de actividades didácticas	 Historia	CURSO	FINALIZADA	J	-	
<a href="#">LAB-58</a>	Definir como implementar e incentivar la creatividad e...	 Historia	CURSO	FINALIZADA	J	-	

Se ha seleccionado cuidadosamente los temas de nuestro curso, cubriendo todo, desde los conceptos básicos de la programación web (como HTML, CSS y Javascript) hasta las funciones más avanzadas de Angular, incorporando TypeScript como requisito. Además, también se incorporó NodeJS + Express para cubrir la programación del lado del servidor y una breve descripción de PostgreSQL alojado en contenedores en Docker. La programación fue una consideración importante, y se organizó el contenido para que encajen perfectamente dentro de las semanas asignadas del calendario de un semestre universitario. Así mismo, se ha explorado la web para compilar recursos complementarios para enriquecer las lecciones teóricas que se presentan en el curso, incluidos videos, enlaces relevantes, material de lectura y ejercicios prácticos. El

objetivo es asegurar que los estudiantes que tomen este curso obtengan experiencia práctica que sirva para el desarrollo de retos académicos y profesionales.

**Figura 38.**

*Planificación Curso programación en la web con Angular en Meiweb*

The screenshot displays the MeiWeb interface for course planning. The header includes the MeiWeb logo, navigation links (Inicio, Atrás, Ayuda, Salir), and university information (Universidad Industrial de Santander, JAVIER FERNANDO CARVAJAL SANABRIA, Lunes, 25 de Septiembre de 2023).

The main content area is titled "Ver Planificador de actividades PCWEB\_ANGULAR\_LABCLINICO". It shows the following details:

- Semana: 1**
- Docente: JAVIER FERNANDO CARVAJAL SANABRIA**
- Finalizar** (button)
- Eliminar** (button)
- Fecha inicio:** 2023-07-10
- Fecha finalización:** 2023-07-16

The interface is divided into sections for course planning:

- COMPETENCIAS CONCEPTUALES - SABER CONOCER**
  - PRESENTACIÓN DEL CURSO**
    - Presentación de estudiantes y profesor
    - Planificación de la asignatura
    - Definición de evaluación y Fechas
    - Demostración y práctica del portal educativo meiweb 9
    - Introducción a la programación web e Internet
    - Conceptuar parte de la historia de la programación web
  - Laboratorio**
    - Instalación de Herramientas software para web
    - Prácticas con la plataforma Meiweb9
- COMPETENCIAS PROCEDIMENTALES - SABER HACER**
  - Competencias para:**
    - Conocer el contenido de la asignatura
    - Establecer las políticas del curso
    - Cronograma de evaluaciones y Estrategias pedagógicas
    - Practica en la plataforma de aula virtual [meiweb9.uis.edu.co](http://meiweb9.uis.edu.co); es un LMS (Learn Management System) desarrollado en la EISI y dirigida por el profesor Manuel Guillermo Flórez Becerra
    - Realizar una Introducción a la Web, conocer terminología Web
- ACTIVIDADES**
  - Actividades en clase**

The left sidebar contains navigation links: Materias, Contenidos, Actividades, Evaluaciones, Banco de Preguntas, Calificaciones, Herramientas, Glosario, FAQ, SCORM, and Lanzadores. The right sidebar contains user links: Mi Perfil, Listas, Correo, Foro, Chat, Cartelera, Calendario, and Estadística.

**Figura 39.**

*Planificación Curso programación en la web con Angular en Moodle*



Al momento de crear el curso en MeiWeb y Moodle, se evaluaron las siguientes competencias: competencias procedimentales y competencias conceptuales. El fin de plantear estas competencias es incentivar un pensamiento creativo en el proceso de aprendizaje del estudiante que permita la solución de problemas y retos.



**Figura 40.**

*Contenido curso programación en la web*

Volver 16. Semana 14 Exportar

### Login con passport.js y jwt

- Documentación oficial de Passport.js: <http://www.passportjs.org/docs/>
- Estrategia local de passport.js: <http://www.passportjs.org/packages/passport-local/>
- Cómo hacer un login con Passport y Node.js - Tutorial: <https://www.youtube.com/watch?v=QyKWhMy2eNo>
- Documentación oficial de Passport.js sobre las estrategias disponibles: <http://www.passportjs.org/docs/strategies/>
- NodeMailer documentación: <https://nodemailer.com/about/>
- ¿Qué es el token web JSON?: <https://jwt.io/introduction/>
- Autenticación sencilla basada en tokens: <https://auth0.com/es/learn/token-based-authentication-made-easy>
- Documentación oficial de jsonwebtoken (la biblioteca que se usa para manejar JWT en Node.js): <https://www.npmjs.com/package/jsonwebtoken>
- Documentación oficial de Passport.js sobre la estrategia de autenticación JWT: <http://www.passportjs.org/packages/passport-jwt/>
- Cómo implementar la autenticación API con JSON Web Tokens y Passport: <https://www.digitalocean.com/community/tutorials/api-authentication-with-json-web-tokens-jwt-and-passport>

#### Cómo hacer un login con Passport y Node.js



Adicional, cada semana tenía una hoja de ruta para la clase, esto es de gran ayuda para el docente y para el estudiante ya que permite tener una guía de lo que se quiere tratar en cada clase. El documento con la ruta de clase se puede encontrar en la semana correspondiente en el Meiwweb y Moodle.

**Figura 41.**

*Hoja de ruta para las clases de cada semana*

# HTML Y CSS

semana 1

CLASE I:

Subtemas y recursos:

- Introducción a HTML para principiantes:  
<https://www.w3schools.com/html/default.asp>
- ¿Qué es HTML y para qué se utiliza? Una introducción:  
[https://developer.mozilla.org/es/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/HTML_basics)
- Estructura HTML  
<https://blog.hubspot.es/website/como-hacer-una-pagina-web-en-html#:~:text=La%20estructura%20b%C3%A1sica%20de%20una,visibles%20dentro%20de%20la%20p%C3%A1gina.>
- Estilos con HTML  
[https://www.w3schools.com/html/html\\_styles.asp](https://www.w3schools.com/html/html_styles.asp)
- Imágenes en HTML  
[https://www.w3schools.com/html/html\\_images.asp](https://www.w3schools.com/html/html_images.asp)
- Bloques en HTML  
[https://www.w3schools.com/html/html\\_blocks.asp](https://www.w3schools.com/html/html_blocks.asp)
- Etiquetas de interacción  
<https://developer.mozilla.org/es/docs/Web/HTML/Element/div>
- Etiquetas de interacción  
<https://www.ediciones-eni.com/open/mediabook.aspx?idR=fef5bcfec475dc2a57c42ff29c809060>

Con el fin de cumplir y desarrollar las competencias y objetivos planteados, se hizo uso de herramientas Miro y Padlet que permiten el desarrollo de tareas colaborativas con recursos interesantes que ayudan al estudiante a salir del ambiente común de un salón de clases, esto también permite al docente acompañar y guiar al estudiante en todo el proceso. Para incentivar una competencia sana entre compañeros y un mayor interés en el contenido de clases se hacen cuestionarios por medio de la plataforma Kahoot y se otorgan pequeñas bonificaciones que ya el docente define. Con todo lo anterior se busca involucrar a los estudiantes en experiencias prácticas y actividades concretas para promover un aprendizaje más profundo y significativo (Kolb, D., 2014).

### 6.3 Fase de Review

En esta fase se evalúa el desarrollo del aplicativo y el curso dirigido a los estudiantes, por lo tanto, se divide en 2 periodos de evaluación, uno para cada caso.


#### 6.3.1 Aplicación laboratorio clínico






Con el fin de validar el cumplimiento de los requerimientos establecidos y el correcto funcionamiento del software, se planteó un plan de pruebas para valorar las funcionalidades claves planteadas al inicio del proyecto.





##### 6.3.1.1 Plan de pruebas. Plan de pruebas para el aplicativo.




**Tabla 13.**

*Plan de pruebas*

ID Prueba	Descripción	Entrada	Resultado Esperado	Resultado Obtenido	Estado
RU01	Registrar usuario desde el inicio de la aplicación	Datos válidos para el registro de usuario	Redirige a inicio de sesión	Se registra el usuario, mensaje de confirmación y redirige al inicio de la aplicación	

RU02	Registrar usuario incorrectamente desde el inicio de la aplicación (Datos ingresados no cumplen requerimientos)	Datos inválidos y/o incompletos para el registro de usuario	Mensaje de error por datos inválidos	Mensaje de error y falla el registro	
IS01	Iniciar sesión (Datos correctos)	Correo y contraseña correcta	Redirige a los resultados del usuario	El usuario accede a la sección de resultados correspondientes	
IS02	Iniciar sesión (Datos incorrectos)	Correo y contraseña incorrecta	Mensaje de error por datos inválidos	Falla el inicio de sesión y solicita datos correctos	
RAC01	Recuperar y actualizar contraseña	Correo de usuario válido	Se envía un mensaje por correo para recuperar contraseña	Mensaje de confirmación y correo para realizar el procedimiento	
VR01	Visualizar resultados del usuario	Inicio de sesión válido	Se muestran los resultados clínicos del usuario	Se visualiza el dashboard con todos los resultados correspondientes	

VR02	Visualizar y validar a detalle resultados del usuario	Datos de prueba	Los resultados mostrados son correctos y coinciden con los datos esperados	Se visualiza el resultado deseado a detalle con la información clínica	
CR01	Cargar resultado (No disponible para rol de paciente)	Identificador de paciente y archivo en formato PDF del resultado	Se carga el archivo correctamente y se dispone al usuario correspondiente	Mensaje de confirmación y se disponibiliza al paciente correspondiente	
NP01	Notificar paciente sobre la creación de nuevo resultado clínico a su nombre	Creación de resultado exitosa	Se envía un mensaje por correo al paciente donde se notifica la creación del resultado y los pasos para visualizarlo	Mensaje de confirmación y correo para notificar al paciente sobre el nuevo resultado	
CR02	Cargar resultado con formato inválido	Identificador de paciente y archivo en formato inválido del resultado	Mensaje de error por formato incorrecto	Se Carga el resultado y se procede con la notificación	

CE01	Crear examen nuevo (Disponible solo para rol administrador)	Datos requeridos para un examen	Se crea nuevo examen disponible para resultados	Examen creado y disponible para nuevos resultados	
EE01	Editar examen existente (Disponible solo para rol administrador)	Datos para actualizar examen	Se actualiza el examen existente y para los resultados nuevo se actualizará información	Examen editado e información actualizada para nuevos resultados	
BE01	Borrar Examen existente (Disponible solo para rol administrador)	Examen a eliminar	Se elimina el examen existente y no se va a encontrar disponible para la creación de nuevos resultados desde el momento de la eliminación	Examen eliminado exitosamente	

Se logro llevar a cabo la creación de la aplicación web para un laboratorio clínico. Se tenía como objetivo la muestra de resultados de exámenes para los pacientes, esa era la funcionalidad principal, el aplicativo fue implementado con Angular, Bootstrap, NodeJS, Express y PostgreSQL.

### ***6.3.2 Curso programación en la web***

Para el Review del curso, el profesor de la asignatura de programación web en la Universidad Industrial de Santander ha valorado la estructuración del curso, que los temas del curso cumplan con lo establecido y puedan ser abordado en las semanas del calendario semestral, que el material del curso esté relacionado al aplicativo desarrollado y cumplan con los objetivos planteados.

Se estructuro de manera exitosa el curso de programación en la web, atendiendo las necesidades de la materia tratando temas de (Front-end – Back-end - Base de datos), Se planificaron actividades para que los estudiantes en el transcurso del semestre construyan un aplicativo similar o igual al del laboratorio clínico.

## **7 Conclusiones**

Se culmina el proyecto de manera exitosa cumpliendo con el objetivo general que era diseñar e implementar un curso de aprendizaje para la asignatura “Programación en la web”, usando como prototipo el aplicativo web para la muestra de resultados de laboratorio clínico, cargando su contenido en las dos plataformas de enseñanza virtual Meiweb y Moodle.

Con base en el primer objetivo específico se logró diseñar el curso en las dos plataformas de aprendizaje nombradas anteriormente, de esta manera se logró desarrollar el aplicativo web con su funcionalidad principal, al final de cada sección del curso se muestran ejemplos ilustrativos con base en lo que se trabajó con el aplicativo, la idea principal es que cada estudiante durante el semestre vaya trabajando en la creación y construcción de un aplicativo web similar al ofrecido por el curso, para que el estudiante pueda lograr esto cada semana tendrá que realizar unas actividades que lo guiaran paso a paso al objetivo, finalmente se trabajó para desarrollar

competencias a los estudiantes para incentivar el pensamiento crítico con base en el desarrollo del curso y sus actividades propuestas.

Para culminar se alcanzaron todos los objetivos propuestos al inicio del proyecto y con base en lo propuesto por nosotros esperamos que sea de gran utilidad para la materia programación en la web y pueda generar ese gran impacto de inspiración en los estudiantes.

## **8 Recomendaciones**

- Con base en el curso recomendaría desarrollar nuevas actividades relacionadas con los temas, también sería bueno agregar más tecnologías diferentes a las utilizadas e ir actualizado los contenidos ya que muchas veces salen nuevas versiones en los frameworks.
- Para el aplicativo sería muy bueno agregar más funcionalidades e intentar optimizar todo el proceso entre el cliente y el laboratorio, para que de esta manera se elimine tanta operatividad y todo el proceso quede de forma digital.



### Referencias bibliográficas

- Agiliza tu desarrollo en Nodejs con el ORM sequelize. (s.f.). EDteam - En español nadie te explica mejor. <https://ed.team/blog/agiliza-tu-desarrollo-en-nodejs-con-el-orm-sequelize/>
- Atlassian. (s.f.). Arquitectura de microservicios | Atlassian. <https://www.atlassian.com/es/microservices/microservices-architecture>
- Atlassian. (s.f.). Scrum: que es, cómo funciona y cómo empezar | Atlassian. <https://www.atlassian.com/es/agile/scrum>
- Borges, S., & Borges, S. (2019). Servidor PostgreSQL. Infranetworking. <https://blog.infranetworking.com/servidor-postgresql/>
- Calle, N. R. (2020, 5 Abril). Patrón Arquitectura de Microservicios. Refactorizando. <https://refactorizando.com/arquitectura-de-microservicios/>
- Castellanos, E. (2021). Git vs GitHub – ¿Que es el control de versiones y cómo funciona? freeCodeCamp.org. <https://www.freecodecamp.org/espanol/news/git-vs-github-what-is-version-control-and-how-does-it-work/>
- Conexión Esan. (2017, 1 Junio). ¿Que es el Scrum y por qué es importante desarrollarlo en un proyecto? <https://www.esan.edu.pe/conexion-esan/que-es-el-scrum-y-por-que-es-importante-desarrollarlo-en-un-proyecto#:~:text=Scrum%20es%20%C3%BAtil%20para%20desarrollar,la%20flexibilidad%20y%20la%20productividad.>
- Equipo editorial de IONOS. (2022). ¿Que es Jira? Todo lo que necesitas saber sobre el software de gestión de proyectos. IONOS Digital Guide. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-jira/>
- Fantino, J. (2021, 22 Enero). ¿Que es el desarrollo web?: Guía para detectar el desarrollador web que vive en ti. <https://www.crehana.com/blog/transformacion-digital/que-es-desarrollo-web/>
- Fit Learning. (2022). Tendencias de e-learning en Colombia. <https://fitls.com/blog/tendencias-de-e-learning-en-colombia/>
- Huet, P. (2023, 13 abril). Arquitectura de software: que es y que tipos existen. OpenWebinars.net. <https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>

- Kinsta. (2022, 19 diciembre). ¿Que es Express.js? todo lo que debes saber. Kinsta®. <https://kinsta.com/es/base-de-conocimiento/que-es-express/>
- Kinsta. (2023, 13 enero). ¿Que es PostgreSQL? Kinsta®. <https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/>
- Kolb, D. (2014). Experiential learning: Experience as the source of learning and development. <https://ci.nii.ac.jp/ncid/BB1767575X>
- Lucas, J. (2019, 4 Septiembre). Que es NodeJS y para que sirve. <https://openwebinars.net/blog/que-es-nodejs/>
- Martinekuan. (s.f.). Datos no relacionales y NoSQL - Azure Architecture Center. Microsoft Learn.<https://learn.microsoft.com/es-es/azure/architecture/data-guide/big-data/non-relational-data>
- Muente, G. (2020). Guía completa del framework: que es, cuáles tipos existen y por que es importante en Internet. Rock Content - ES. <https://rockcontent.com/es/blog/framework/>
- Muente, G. (2020, 19 Junio). LMS: lo que necesitas saber de una plataforma de educación virtual. <https://rockcontent.com/es/blog/plataforma-lms/>
- Nestrategia. ¿Que es Back End, Front End y Back Office y por que es importante para tu web? <https://nestrategia.com/desarrollo-web-back-end-front-end/>
- Quality Devs. (2019, 16 septiembre). ¿Que es angular y para que sirve? <https://www.qualitydevs.com/2019/09/16/que-es-angular-y-para-que-sirve/>
- Que es SCRUM. (2021, 20 septiembre). Proyectos Ágiles. <https://proyectosagiles.org/que-es-scrum/>
- ¿Que es la arquitectura de tres niveles? | IBM. (s.f.). <https://www.ibm.com/mx-es/topics/three-tier-architecture>
- ¿Que es una base de datos? (s.f.). <https://www.oracle.com/co/database/what-is-database/>
- ¿Que es el e-learning y cómo está transformando la educación? (2022, 30 Marzo). Santander Universidades. <https://www.becas-santander.com/es/blog/e-learning.html>
- ¿Que es el desarrollo web? (Tekla, 2022, 18 Marzo). <https://tekla.io/blog/que-es-desarrollo-web/>
- ¿Que es el desarrollo web? (Seguro, N., 2021, 17 Septiembre). <https://www.coderhouse.com.pe/blog/que-es-el-desarrollo-web>

¿Que es Express y para que sirve? (Rodas, G., 2018, 12 Noviembre).

<https://platzi.com/clases/1437-express-js/15697-que-es-express-y-para-que-sirve/>

¿Que son las pruebas de integración? (K. Team, 2023, 18 abril). KeepCoding Bootcamps.

<https://keepcoding.io/blog/que-son-las-pruebas-de-integracion/>