

Nueva versión del proyecto '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'.

Yezith Fernando Rincón Guevara y Santiago Ariza Briceño

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Manuel Guillermo Florez Becerra

Msc. En informática

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2025

Agradecimientos

Quiero expresar mi más sincero agradecimiento a mi familia, especialmente a mis padres, por su apoyo constante, su paciencia y su aliento durante toda esta etapa de mi vida académica. Sin su amor y comprensión, este logro no habría sido posible.

A mis compañeros y amigos, en especial a Santiago Ariza, quienes compartieron este camino conmigo, brindándome apoyo, ideas y una motivación constante. Gracias por los momentos de colaboración y convivencia, que hicieron esta experiencia aún más enriquecedora.

Agradezco a mi director de proyecto, Manuel Guillermo, por su orientación y su paciencia. Sus sugerencias fueron esenciales para la culminación exitosa de este proyecto.

En conclusión, este proyecto ha sido posible gracias al apoyo de muchas personas que han hecho parte de mi vida académica y me han ayudado. A todos ellos, mi más profundo agradecimiento.

Agradecimientos

A Deya, mi madre, por brindarme todo tu amor sin condiciones y ser mi ejemplo de fortaleza. Sin cada uno de los sacrificios que hiciste por mí, no habría manera alguna de lograrlo. Esto es por ti, para ti.

A Javier, mi padre, por sostener cada uno de mis pasos y enseñarme la importancia de luchar. Gracias porque, al tenerte como ejemplo de vida, me di cuenta de que no hay obstáculo que no pueda vencer.

A Nicolás, mi hermano, por ser siempre mi compañía en la vida. Siempre hemos estado los dos contra todo, esto también es gracias a ti.

A Monroy e Iván, mis amigos, por estar a mi lado incondicionalmente y porque cada momento en el que dedicamos nuestro tiempo a disfrutar de la vida, es combustible que me permite continuar.

A Yezith, mi compañero de universidad y amigo, por haber compartido conmigo el difícil camino que fue llegar hasta aquí.

A Nala, mi ángel.

Finalmente, a mis familiares, en especial a mi tía Aive por tanto apoyo y por haberme regalado alguna vez el computador que me permitió seguir estudiando y desde el que hoy le escribo gracias. A mi tía Victoria por toda la ayuda y las palabras de aliento recibidas a lo largo de mi etapa universitaria. Y a mi tía Mónica, por ser un pilar de apoyo para mi familia y por estar siempre presente para nosotros.

Contenidos

Introducción.....	14
1. Planteamiento y justificación del problema.....	15
2. Objetivos.....	16
2.1 Objetivo general	16
2.2 Objetivos específicos	16
3. Marco teórico.....	17
3.1 Marco de referencia conceptual.....	17
3.1.1 Desarrollo web.....	17
3.1.2 Historia y evolución de la programación en la web	17
3.1.3 Arquitectura de software	18
3.1.3.1 Modelo de capas	19
3.1.3.2 Patrón cliente – servidor.....	19
3.1.3.3 Modelo de desarrollo MVC (Modelo-Vista-Controlador)	20
3.1.4 Frameworks	21
3.1.5 Front-end	21
3.1.5.1 Bootstrap.....	22
3.1.5.2 Angular y Typescript.....	22
3.1.6 Backend	23
3.1.6.1 NodeJS.....	24

3.1.6.2 Express	25
3.1.7 Seguridad web.....	25
3.1.8 Pruebas y depuración.....	25
3.1.9 Inteligencia Artificial	26
3.1.9.1 CHATGPT	27
3.1.10 Gestión de proyectos y metodologías de desarrollo	27
3.1.10.1 SCRUM	27
3.1.10.2 Bases de datos.....	28
3.1.11 Bases de datos relacionales.....	28
3.1.11.1 PostgreSQL.....	28
3.1.12 Bases de datos no relacionales.....	29
3.1.13 Aprendizaje en línea o E-learning	29
3.1.14 Aprendizaje en línea de tipo B-learning	29
3.1.14.1 Meiweb	30
3.1.14.2 Moodle.....	30
4. Estado del arte	31
4.1 Antecedentes de la situación de estudio	31
5. Metodología.....	32
5.1 Planning	32
5.2 Desarrollo	33

5.3 Revisión 33

6. Desarrollo de proyecto..... 34

6.1 Fase de planning 34

6.1.1 Herramientas 34

6.1.1.1 Angular 34

6.1.1.2 Bootstrap..... 34

6.1.1.3 NodeJS y Express 34

6.1.1.4 PostgreSQL..... 35

6.1.2 Control de tareas 35

6.1.2.1 Trello..... 35

6.1.3 Control de versiones 35

6.1.3.1 Git y Github 35

6.1.4 Despliegue 35

6.1.4.1 Render..... 36

6.1.5 Pruebas..... 36

6.1.5.1 Cypress 37

6.1.6 Análisis de Desarrollo 37

6.1.6.1 Requerimientos funcionales 38

6.1.6.2 Requerimientos no funcionales 41

6.1.6.3 Casos de uso	44
6.1.6.4 Diagrama de actividades.....	45
6.1.6.5 Arquitectura	47
6.1.7 Interfaz de usuario	48
6.2 Fase de desarrollo	49
6.2.1 Aplicativo web clínico.....	49
6.2.1.1 Desarrollo backend.....	50
6.2.1.2 Desarrollo frontend.....	53
6.2.2 Curso Programación en la Web.....	59
6.2.2.1 Contenido del curso	61
6.2.2.2 Planificación del curso	62
6.2.3 Guía básica de la metodología Scrum	62
6.3 Fase de evaluación.....	64
6.3.1 Aplicativo web clínico.....	64
6.3.1.1 Plan de pruebas.....	64
6.3.2 Curso Programación en la Web.....	68
6.3.3 Guía básica de la metodología Scrum	68
7. Resultados.....	69
7.1 Aplicativo web clínico.....	69

7.2 Curso Programación en la Web	69
7.3 Guía básica de la metodología Scrum	69
8. Conclusiones.....	70
9. Recomendaciones	71
Referencias Bibliográficas.....	72
Apéndices	74

Lista de tablas

Tabla 1. Requerimiento funcional 1..... 38

Tabla 2. Requerimiento funcional 2..... 38

Tabla 3. Requerimiento funcional 3..... 39

Tabla 4. Requerimiento funcional 4..... 39

Tabla 5. Requerimiento funcional 5..... 40

Tabla 6. Requerimiento funcional 6..... 40

Tabla 7. Requerimiento funcional 7..... 41

Tabla 8. Requerimiento no funcional 1..... 41

Tabla 9. Requerimiento no funcional 2..... 42

Tabla 10. Requerimiento no funcional 3..... 42

Tabla 11. Requerimiento no funcional 4..... 43

Tabla 12. Requerimiento no funcional 5..... 43

Tabla 13. Requerimiento no funcional 6..... 44

Tabla 14. Plan de pruebas 64

Lista de figuras

Figura 1. Modelo de capas 19

Figura 2. Modelo cliente servidor..... 20

Figura 3. Modelo vista controlador..... 21

Figura 4. Actores en el sistema 37

Figura 5. Diagrama de casos de uso..... 44

Figura 6. Diagrama de inicio de sesión..... 45

Figura 7. Diagrama para registro creación de cuenta 45

Figura 8. Diagrama para ver resultados 46

Figura 9. Diagrama para subir resultados 46

Figura 10. Diagrama para crear un examen 47

Figura 11. Arquitectura rest 48

Figura 12. Mockup inicio de sesión..... 48

Figura 13. Tablero de tareas..... 49

Figura 14. Lista de tareas aplicativo clínico 50

Figura 15. Arquitectura api-rest..... 50

Figura 16. Estructura de carpetas backend 51

Figura 17. Estructura de carpetas frontend 53

Figura 18. Estructura de carpetas frontend carpeta src 54

Figura 19. Vista inicio de sesión..... 55

Figura 20. Vista registrarse 55

Figura 21. Vista inicial home..... 56

Figura 22. Vista de servicios..... 56

Figura 23. Vista resultados por paciente.....	56
Figura 24. Vista subir resultados	57
Figura 25. Vista crud exámenes.....	57
Figura 26. Vista modal para creación y edición de exámenes	57
Figura 27. Vista gestión de usuarios	58
Figura 28. Vista modal para la gestión/edición de información de usuarios	58
Figura 29. Vista página de contacto.....	58
Figura 30. Vista página no encontrada o 404	59
Figura 31. Lista de tareas para creación del curso	60
Figura 32. Planificador en Moodle del curso Programación en la Web con Angular	60
Figura 33. Planificador en Meiweb del curso Programación en la Web con Angular.....	61
Figura 34. Contenido del curso semana 1 Programación en la Web con Angular	61
Figura 35. Ejemplo Actividades semana 2 en planificador	62
Figura 36. Primera página de la Guía básica de la metodología Scrum	63
Figura 37. Primera página del formato de plan de proyecto con Scrum.....	63

Resumen

Título: Nueva versión del proyecto '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:** Yezith Fernando Rincón Guevara y Santiago Ariza Briceño**

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

Descripción: Se abordan algunos temas clave para el desarrollo del proyecto, tales como: conceptos de programación web, introducción a nuevas tecnologías para el desarrollo web, la inclusión de inteligencias artificiales en el proceso educativo y otros conceptos que faciliten la comprensión y ejecución del proyecto, todo esto respaldado por fundamentos teóricos y contenidos similares a los propuestos. Para lograr esto, se establece una hoja de ruta formal que incluye la presentación y justificación del problema, los objetivos planteados, el cronograma, la metodología y el presupuesto necesario para la implementación del proyecto. Además, se detalla el desarrollo del mismo mostrando los requisitos, diagramas, entre otros aspectos. Finalmente, se describen las pruebas realizadas, mostrando la funcionalidad de la página y el contenido de los cursos creados para las plataformas de aprendizaje Moodle y MeiWeb. Se concluye con un análisis basado en los objetivos generales y específicos planteados al inicio del documento, y se dan recomendaciones para trabajos futuros y posibles mejoras que permitan generar un mayor impacto en los estudiantes de la asignatura de 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

Abstract

Title: New version of the project 'Course model for the subject “Web Programming” using the Angular framework and implementation of a web application for a clinical laboratory'.*

Authors: Yezith Fernando Rincón Guevara y Santiago Ariza Briceño**

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

Description: Some key topics for the development of the project are addressed, such as: web programming concepts, introduction to new technologies for web development, the inclusion of artificial intelligences in the educational process and other concepts that facilitate the understanding and implementation of the project, all supported by theoretical foundations and contents similar to those proposed. To achieve this, a formal roadmap is established that includes the presentation and justification of the problem, the proposed objectives, the chronogram, the methodology and the necessary budget for the implementation of the project. In addition, the development of the project is detailed, showing the requirements, diagrams, among other aspects. Finally, the tests performed are described, showing the functionality of the page and the content of the courses created for the Moodle and MeiWeb learning platforms. It concludes with an analysis based on the general and specific objectives stated at the beginning of the document, and provides recommendations for future work and possible improvements to generate a greater impact on the students of the web programming course.

* Degree Project. Research Work.

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

Engineering. Director: Msc. En informática Manuel Guillermo Florez Becerra

Introducción

Hoy en día, es crucial aprovechar al máximo los beneficios que ofrece Internet. Por esta razón, las aplicaciones web se han vuelto herramientas indispensables para empresas y comercios. No solo permiten llegar a más clientes y automatizar procesos comerciales, sino que también representan la imagen de las empresas ante los usuarios, influyendo significativamente en su percepción de la marca y decisión de compra.

A pesar de que existen diversos lenguajes de programación enfocados en el desarrollo web, Javascript es el más popular entre los desarrolladores debido a su flexibilidad y amplias funcionalidades, lo que lo convierte en un lenguaje potente. A medida que su popularidad crece, surgen nuevas tecnologías que facilitan el desarrollo y ofrecen nuevas capacidades al desarrollador, como Angular y NodeJs, que permiten realizar tanto el front-end como el back-end de los sitios y las aplicaciones web con JavaScript.

Este proyecto buscó validar y mejorar el curso virtual de aprendizaje sobre Angular y NodeJS que se creó para la asignatura de programación en la web en las plataformas Meiweb y Moodle. También se añadieron nuevas funcionalidades al prototipo 'laboratorio clínico' que se desarrolló como ejemplo central del curso virtual con el objetivo de lograr una versión más completa del aplicativo.

El proyecto se llevó a cabo utilizando una metodología basada en el marco de trabajo SCRUM. Inicialmente, estuvo la fase de planificación, en la que se analizaron los requerimientos y, a partir de esto, se organizaron las etapas siguientes. Luego, siguió la fase de desarrollo, durante la cual se buscó la implementación del prototipo. Finalmente, en la fase de revisión, mediante retroalimentación, se evaluó el progreso alcanzado.

1. Planteamiento y justificación del problema

Las tecnologías web desempeñan un papel fundamental en la sociedad de hoy, ya que han transformado la forma en que interactuamos, compartimos información y realizamos negocios. Desde los inicios de la internet, estas tecnologías han evolucionado continuamente, abarcando una amplia gama de aplicaciones que van desde sitios estáticos hasta complejas plataformas interactivas y servicios en la nube.

Con el alcance cada vez mayor de las tecnologías web, su aprendizaje se ha convertido en una parte vital de la formación de los ingenieros de sistemas que deciden dedicarse al desarrollo web. En este contexto, se hace visible la importancia de un curso de programación en la web adecuado, que esté actualizado y correctamente estructurado. Es por esto por lo que se quiere validar el modelo de curso virtual que se realizó anteriormente en las plataformas Meiweb y Moodle, para posteriormente actualizar, reorganizar y mejorar los contenidos de dicho curso, además de establecer un cronograma de tal manera que se sintetizen los conocimientos esenciales de la programación web en el corto tiempo del que se dispone durante el semestre. Por último, se busca modificar el aplicativo de laboratorio clínico que se toma de ejemplo para el desarrollo del curso, añadiendo nuevas funcionalidades para lograr una versión completa del aplicativo.

2. Objetivos

2.1 Objetivo general

- Validar y mejorar la versión anterior del curso virtual de Angular y NodeJS en las plataformas de enseñanza Meiweb y Moodle para la asignatura de Programación en la Web.

2.2 Objetivos específicos

- Actualizar el contenido del curso virtual de Angular y NodeJs en las plataformas Meiweb y Moodle.
- Ampliar y mejorar el aplicativo de laboratorio clínico realizado como ejemplo de proyecto del curso virtual, integrando nuevas funcionalidades.
- Establecer el cronograma de planificación del curso virtual en las plataformas Meiweb y Moodle.
- Crear una guía básica de la metodología Scrum para apoyar el desarrollo del proyecto final de la asignatura.

3. Marco teórico

3.1 Marco de referencia conceptual

3.1.1 Desarrollo web

El desarrollo web es el proceso de creación, diseño y mantenimiento de sitios web y aplicaciones web, abarcando una amplia gama de actividades que van desde la planificación inicial hasta la implementación y la actualización continua. Implica el uso de diferentes tecnologías como HTML, CSS y JavaScript para estructurar y diseñar la interfaz de usuario, así como el empleo de diversos lenguajes de programación del lado del servidor, como PHP, Python o Node.js, para gestionar la lógica y la funcionalidad del sitio. Además, el desarrollo web incluye aspectos relacionados con la optimización para motores de búsqueda (SEO), la accesibilidad, la seguridad y la experiencia del usuario, con el objetivo de crear sitios web eficientes, atractivos y fáciles de usar que satisfagan las necesidades y expectativas de los usuarios en línea.

3.1.2 Historia y evolución de la programación en la web

El desarrollo web tiene sus raíces en la creación de tres elementos fundamentales: el lenguaje de marcado HTML (HyperText Markup Language), el protocolo de transferencia de hipertexto HTTP (Hypertext Transfer Protocol) y el sistema de localización de objetos en la web, conocido como URL (Uniform Resource Locator).

HTML, que fue desarrollado por *Tim Berners-Lee* en 1990, proporcionó la estructura básica para la creación de páginas web al permitir la organización y presentación de contenido mediante etiquetas y atributos. El protocolo HTTP, también concebido por Berners-Lee, se introdujo en 1991 y estableció las reglas para la comunicación entre servidores web y clientes, facilitando la transferencia de información en la World Wide Web. Por último, las URL, que surgieron en 1994, proporcionaron un sistema de direccionamiento uniforme para localizar recursos en la web, permitiendo a los usuarios acceder a páginas web específicas mediante direcciones únicas y legibles para humanos (Wikipedia, 2024).

En 1995, la introducción de JavaScript comenzó a abordar las limitaciones de HTML, marcando un punto de inflexión en el dinamismo del diseño web. Posteriormente, la llegada de Macromedia Flash ofreció a los diseñadores una variedad de opciones creativas y animadas.

El año 1998 presenció la aparición tanto de PHP como de CSS, permitiendo la creación de sitios web más dinámicos y reduciendo los tiempos de carga al eliminar la necesidad de utilizar tablas en el diseño web, lo que posibilitó realizar más con menos código.

A medida que la velocidad de Internet aumentaba en los primeros años del 2000, se intensificó el interés por soluciones innovadoras en software y hardware para el diseño web.

El año 2003 marcó un hito con el surgimiento de Facebook y WordPress, que transformaron radicalmente el panorama de las redes sociales y la creación de sitios web, respectivamente, este último llegando a impulsar más del 35% de los sitios en línea para el 2020.

En el 2010, la proliferación de smartphones introdujo el concepto de diseño web responsive. A partir de entonces, los diseñadores contaron con herramientas de alta calidad y una amplia gama de software y hardware, lo que les permitió refinar el estilo de los sitios web como nunca antes (Castelán, 2022).

3.1.3 Arquitectura de software

Independientemente de los lenguajes de programación o de las tecnologías que se planeen utilizar, antes del proceso de desarrollo se consideran muchos aspectos relacionados con el comportamiento esperado del software. Cosas como los requerimientos de usuarios o peticiones simultáneas, la velocidad de carga del sitio, la disponibilidad o la accesibilidad van a determinar varias de las decisiones de diseño que definirán el modo en que se desarrollará el proyecto.

Cuando se tienen proyectos muy grandes y/o complejos, se deben utilizar herramientas de organización y abstracción que permitan controlar como y donde encajan cada uno de los componentes de la aplicación. Entre estas herramientas se encuentran los patrones de diseño que ayudan a abordar

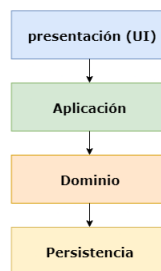
ciertos problemas específicos a nivel del código de una manera ordenada y sencilla, y también los patrones de arquitectura que de manera similar solucionan ciertos problemas comunes, pero teniendo un alcance más amplio, estos patrones arquitectónicos nos indican como está conformada la aplicación. A continuación, se ilustran algunos de los patrones más usados en el diseño de sitios y aplicaciones web.

3.1.3.1 Modelo de capas

Este modelo separa los programas en grupos según las funciones que desempeñan, estos grupos se organizan en capas superpuestas donde cada capa proporciona servicios a la capa superior. Las capas más utilizadas son:

- Capa de presentación (o capa de interfaz de usuario)
- Capa de aplicación (o capa de servicio)
- Capa de lógica de negocios (o capa de dominio)
- Capa de acceso a datos (o capa de persistencia)

Figura 1. Modelo de capas

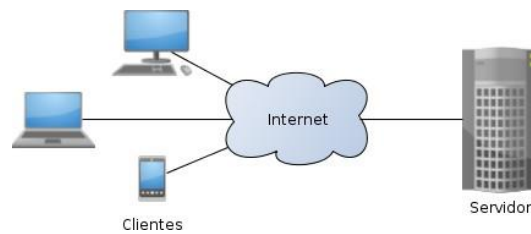


3.1.3.2 Patrón cliente – servidor

El patrón cliente-servidor es un modelo arquitectónico fundamental en el desarrollo de aplicaciones distribuidas, donde dos componentes principales, el cliente y el servidor, interactúan entre sí para proporcionar servicios y recursos. En este patrón, el cliente envía solicitudes al servidor, que a su vez procesa estas solicitudes y envía respuestas al cliente. El cliente es responsable de la presentación y la interacción con el usuario, mientras que el servidor maneja la lógica de negocio,

el procesamiento de datos y el almacenamiento.

Figura 2. Modelo cliente servidor



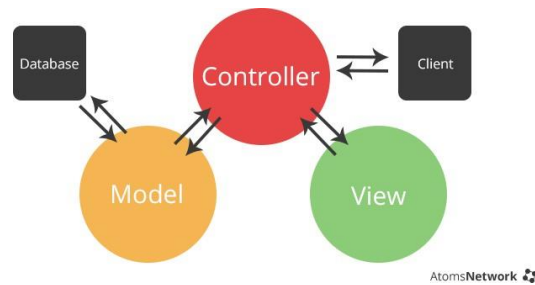
3.1.3.3 Modelo de desarrollo MVC (Modelo-Vista-Controlador)

MVC es una propuesta de arquitectura del software utilizada para separar el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, lo que ofrece beneficios diversos como software más robusto con un ciclo de vida más adecuado, donde se potencie la facilidad de mantenimiento, reutilización del código y la separación de conceptos (Qué es MVC, 2014).

El patrón MVC se constituye de tres componentes principales:

- **Modelo:** Esta capa se encarga de representar los datos y la lógica de negocio de la aplicación. Aquí se definen las estructuras de datos, las reglas de validación y cualquier operación relacionada con la manipulación de la información.
- **Vista:** La vista es la interfaz de usuario que presenta la información al usuario final. Puede ser una página web, una ventana de una aplicación de escritorio o cualquier otro medio de presentación. La vista muestra los datos al usuario y responde a sus interacciones, pero no realiza ningún procesamiento de datos.
- **Controlador:** El controlador actúa como intermediario entre el modelo y la vista. Se encarga de manejar las interacciones del usuario, como hacer clic en un botón o enviar un formulario. Cuando el usuario realiza una acción, el controlador actualiza el modelo según sea necesario y luego actualiza la vista para reflejar los cambios.

Figura 3. Modelo vista controlador



3.1.4 Frameworks

Dado que la mayoría de los proyectos web tienen los mismos problemas y necesidades en común, se han desarrollado los llamados frameworks o marcos de trabajo que proporcionan una guía general para el proceso de desarrollo. Estos frameworks están diseñados en un lenguaje en específico (JavaScript, Python, Java, etc.) y algunos se enfocan en un tipo concreto de aplicación como los que facilitan la creación de las **Single Page Application (SPA)**, suelen además incluir una gran cantidad de utilidades para facilitar la implementación de las características más usadas en las aplicaciones web, como pueden ser el manejo de solicitudes HTTP, la consulta a bases de datos, los sistemas de login de usuarios o el enrutamiento a las distintas secciones dentro del sitio.

3.1.5 Front-end

Front End es la parte de una aplicación que interactúa con los usuarios, es conocida como el lado del cliente. Básicamente es todo lo que vemos en la pantalla cuando accedemos a un sitio web o aplicación: tipos de letra, colores, adaptación para distintas pantallas (RWD), los efectos del ratón, teclado, movimientos, desplazamientos, efectos visuales y otros elementos que permiten navegar dentro de una página web. Este conjunto crea la experiencia del usuario (Comunicación, 2019).

Para lograr crear estas experiencias del usuario se utilizan principalmente tres tecnologías que son los pilares fundamentales de la web de hoy en día: HTML, CSS y JAVASCRIPT. El lenguaje HTML permite, como se mencionó anteriormente, organizar contenidos para mostrarlos en la web por medio de etiquetas y atributos; el lenguaje CSS por su parte, es el que permite agregar estilos a cada uno de los elementos que se tienen en el documento HTML, permite no solo cambiar los

colores, tipos y tamaños de letra sino posicionar todos los elementos en la pantalla y cambiar su tamaño y disposición según el tamaño de las pantallas; hasta aquí todo son elementos estáticos con los que no se puede interactuar, para añadir las funcionalidades y la interacción del usuario entra en juego JavaScript, lenguaje de programación que funciona en los navegadores de internet con el que se puede programar y modificar completamente el comportamiento de los sitios web.

3.1.5.1 Bootstrap

Bootstrap es un framework front-end de código abierto desarrollado por Twitter. Está diseñado para facilitar y agilizar el proceso de desarrollo de sitios web y aplicaciones web responsivas y móviles. Bootstrap proporciona un conjunto de herramientas y componentes preestablecidos, como cuadros de diálogo, botones, formularios, barras de navegación, entre otros, que pueden ser fácilmente integrados en el diseño de una aplicación o sitio web.

Una de las características más destacadas de Bootstrap es su sistema de rejillas (grid system), que permite crear diseños de página flexibles y adaptables a diferentes dispositivos y tamaños de pantalla. Además, Bootstrap incluye un conjunto de estilos CSS prediseñados y componentes JavaScript que ayudan a mejorar la apariencia y la funcionalidad de los sitios web de manera consistente y profesional.

3.1.5.2 Angular y Typescript

Como se mencionó antes, los frameworks son herramientas creadas pensando en agilizar el desarrollo de ciertos proyectos específicos, y estos frameworks están escritos en un lenguaje determinado. En el caso del front end se tienen una cantidad ingente de frameworks, Angular, React, Vue, Svelte, etc., la mayoría de ellos enfocados al desarrollo por componentes y utilizando JavaScript como lenguaje, pues es el rey indiscutible en los navegadores web. Pero también muchos de ellos han empezado a adoptar el lenguaje Typescript y de hecho el framework del que se va a hablar a continuación, Angular, lo usó desde su primera versión.

Typescript es de hecho una extensión del lenguaje JavaScript, desarrollado por Microsoft agregando características de tipado estático opcional y características orientadas a objetos. Esencialmente, TypeScript proporciona una capa de abstracción sobre JavaScript que permite a los desarrolladores escribir código más estructurado y robusto al tiempo que aprovecha todas las funcionalidades de JavaScript. Gracias a esto muchos de los frameworks que se desarrollaron en JavaScript pueden usar Typescript en su lugar y tener así un código más ordenado y legible.

Angular fue uno de los primeros frameworks para crear aplicaciones web dinámicas y de una sola página que salieron al mercado, pues existían algunas herramientas y librerías para crear estas aplicaciones, pero ninguna era tan conveniente como este framework. Fue desarrollado por Google, específicamente por los ingenieros *Misko Hevery* y *Adam Abrons*, y lanzado inicialmente en 2012 bajo el nombre de AngularJS, aunque como se menciona anteriormente este fue desarrollado usando Typescript. Este framework usa la famosa arquitectura modelo-vista-controlador que se usa en muchas aplicaciones web (Deyimar, 2020). Angular proporciona un conjunto completo de herramientas y características que hacen que el desarrollo de aplicaciones web y móviles sea más eficiente, escalable y mantenible, lo que lo convierte en una opción popular para desarrolladores y equipos de desarrollo de todo el mundo.

3.1.6 Backend

"La palabra, en inglés, se refiere a 'lo que está detrás', 'lo que no se ve'. De ahí su término 'back'. Cuando hablamos de backend, hablamos únicamente del código interno de la página" (Gómez, 2022).

Esta parte de atrás, también conocida como el lado del servidor, es la parte de una aplicación o sitio web que se encarga de la lógica y la gestión de los datos detrás de la escena, es decir, todo lo que sucede en el servidor y no es visible para el usuario final. Esta parte del desarrollo web maneja la interacción con la base de datos, la autenticación y autorización de usuarios, la lógica de negocio y cualquier procesamiento que se requiera para la funcionalidad del sistema.

Para la creación y el funcionamiento de estos robustos sistemas se emplean una amplia gama de tecnologías y herramientas. Estas incluyen una variedad de lenguajes de programación, como JavaScript con su popular framework Node.js, Python con Django o Flask, Java con Spring Boot, C# con ASP.NET, y Ruby con Ruby on Rails, entre otros. Además, se emplean diversas bases de datos para almacenar y recuperar datos de manera eficiente, como MySQL, PostgreSQL, MongoDB o SQLite. Los desarrolladores también utilizan frameworks específicos que proporcionan un conjunto de herramientas y bibliotecas predefinidas para simplificar el desarrollo, como Express.js para Node.js, Django y Flask para Python, Spring para Java, y Ruby on Rails para Ruby.

Para gestionar las solicitudes HTTP del cliente y servir las páginas web, se emplean servidores web como Apache o Nginx, y los sistemas de APIs (Interfaces de Programación de Aplicaciones) se utilizan para permitir la comunicación entre diferentes sistemas, siendo las APIs RESTful un enfoque común en el desarrollo del back-end, ya que exponen funcionalidades de la aplicación a través de HTTP.

3.1.6.1 NodeJS

Node.js es un entorno de ejecución de un solo hilo, de código abierto y multiplataforma para crear aplicaciones de red y del lado del servidor rápidas y escalables. Se ejecuta en el motor de ejecución de JavaScript V8, y utiliza una arquitectura de E/S basada en eventos y sin bloqueos, lo que la hace eficiente y adecuada para aplicaciones en tiempo real (Qué es Node.js y por qué debería usarlo, 2021).

Una de las características más destacadas de Node.js es su capacidad para manejar múltiples conexiones de manera eficiente mediante el uso de un solo hilo de ejecución no bloqueante, lo que significa que puede manejar muchas solicitudes simultáneas sin consumir una gran cantidad de recursos del sistema. Esto lo hace especialmente adecuado para aplicaciones web que requieren una alta escalabilidad y rendimiento.

Node.js también cuenta con un amplio ecosistema de módulos y paquetes gracias a su gestor de paquetes npm (Node Package Manager), que es uno de los más grandes y activos en la comunidad de desarrollo de JavaScript. Esto facilita la reutilización de código, la colaboración entre desarrolladores y la integración de bibliotecas y herramientas de terceros en las aplicaciones Node.js.

3.1.6.2 Express

Express.js, a menudo referido simplemente como Express, es el framework de servidor más prominente para Node.js, reconocido por su eficiencia en el desarrollo de aplicaciones web y móviles. El corazón de Express.js es su sistema de middleware, que permite a los desarrolladores ejecutar cualquier código, hacer cambios en la solicitud y la respuesta, finalizar el ciclo de solicitud-respuesta, o llamar a la siguiente función de middleware. Esto significa que con cada solicitud HTTP, Express.js puede ejecutar una serie de funciones middleware en secuencia, lo que proporciona una enorme flexibilidad y potencia para manejar diversas operaciones de backend (Dongee, s.f).

3.1.7 Seguridad web

La seguridad web es una preocupación fundamental en el desarrollo de aplicaciones y sitios en línea. Implica la implementación de medidas para proteger la información sensible y garantizar la integridad de los datos. Esto incluye aspectos como la autenticación de usuarios, la protección contra ataques cibernéticos, el cifrado de datos y la monitorización constante de actividades sospechosas. La importancia de la seguridad web radica en la preservación de la confidencialidad y privacidad de los usuarios, así como en la protección de la reputación y la credibilidad de las empresas. Además, un enfoque proactivo en la seguridad web contribuye a evitar posibles infracciones y a mantener la confianza de los usuarios en las plataformas en línea.

3.1.8 Pruebas y depuración

La depuración es el proceso de encontrar y solucionar errores en el código fuente de cualquier software. Cuando un software no funciona tal y como se espera, los programadores de computadoras estudian el código para determinar por qué ocurren algunos errores. Con este objetivo,

los programadores utilizan herramientas de depuración para ejecutar el software en un entorno controlado, comprobar el código paso a paso y analizar y solucionar el error (Amazon, s.f), 2025).

Por lo tanto, las pruebas y la depuración son aspectos cruciales en el desarrollo de software, ya que garantizan el correcto funcionamiento y la calidad de las aplicaciones. Las pruebas incluyen diversas estrategias, como pruebas unitarias, de integración y de aceptación, que permiten identificar y corregir errores en el código. La depuración requiere un análisis detallado del código y el seguimiento de la ejecución del programa. La realización de pruebas exhaustivas y una depuración eficaz son fundamentales para ofrecer a los usuarios una experiencia óptima y libre de errores.

3.1.9 Inteligencia Artificial

En su forma más simple, la inteligencia artificial es un campo que combina la informática y conjuntos de datos robustos para permitir la resolución de problemas. También engloba los subcampos del aprendizaje automático y el aprendizaje profundo, que se mencionan con frecuencia junto con la inteligencia artificial. Estas disciplinas están compuestas por algoritmos de IA que buscan crear sistemas expertos que hagan predicciones o clasificaciones basadas en datos de entrada (¿Qué es la inteligencia artificial (IA)?, 2024).

La inteligencia artificial (IA) ha revolucionado el desarrollo de software al proporcionar herramientas como el aprendizaje automático y el procesamiento del lenguaje natural. Estas tecnologías permiten a las máquinas realizar tareas que normalmente requerirían inteligencia humana, como el reconocimiento de patrones, la toma de decisiones y la generación de respuestas contextuales. La IA se ha convertido en un componente fundamental en una amplia gama de aplicaciones de software, desde asistentes virtuales hasta sistemas de recomendación y análisis de datos. Su capacidad para automatizar procesos, optimizar sistemas y mejorar la experiencia del usuario la hace invaluable en el desarrollo de software moderno.

3.1.9.1 CHATGPT

ChatGPT, desarrollado por OpenAI, ejemplifica cómo la inteligencia artificial puede aplicarse en el desarrollo de software. Este modelo de lenguaje genera respuestas coherentes y contextualmente relevantes basadas en el texto de entrada. En el desarrollo de software, herramientas como ChatGPT pueden utilizarse para automatizar tareas de atención al cliente, generar contenido relevante, mejorar la interacción usuario-sistema y proporcionar asistencia en la resolución de problemas. Además, al integrar ChatGPT en aplicaciones y sistemas, se puede conseguir mejorar la experiencia del usuario al ofrecer respuestas rápidas y precisas, aumentando así la eficiencia y la satisfacción de este.

3.1.10 Gestión de proyectos y metodologías de desarrollo

La gestión de proyectos y las metodologías de desarrollo juegan un papel crucial en el éxito de cualquier proyecto de software. La correcta planificación, organización y ejecución de las actividades son fundamentales para garantizar la entrega oportuna y eficiente de un producto de software de alta calidad. Las metodologías ágiles, como Scrum, se centran en la flexibilidad, la colaboración y la entrega incremental, lo que permite adaptarse rápidamente a los cambios en los requisitos y prioridades del cliente. Por otro lado, las metodologías tradicionales, como la cascada, siguen un enfoque más estructurado y secuencial, ideal para proyectos con requisitos estables y bien definidos desde el principio. La elección de la metodología adecuada depende del contexto del proyecto, los objetivos del cliente y las preferencias del equipo de desarrollo. Una gestión efectiva del proyecto implica la aplicación de técnicas de planificación, seguimiento y control, así como la comunicación transparente y la colaboración entre todos los miembros del equipo.

3.1.10.1 SCRUM

Scrum es una metodología ágil centrada en entregas iterativas e incrementales. Los proyectos se dividen en sprints cortos, generalmente de 2 a 4 semanas, para desarrollar y entregar un incremento del producto. El equipo, liderado por un Scrum Master, colabora con el cliente para priorizar requisitos en un backlog. Durante cada sprint, el equipo selecciona elementos del backlog, se compromete a completarlos y realiza reuniones diarias para sincronizarse. Al final de cada sprint, se

realiza una revisión y retrospectiva para demostrar el trabajo completado y mejorar el proceso continuamente.

3.1.10.2 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. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS) (¿Qué es una base de datos?, 2020).

Las bases de datos pueden ser relacionales, estructuradas en tablas con filas y columnas, ideales para datos estructurados como nombres y fechas. O no relacionales, flexibles y escalables, adecuadas para grandes volúmenes de datos no estructurados como documentos o datos geoespaciales. Ambos tipos tienen ventajas y se eligen según las necesidades del proyecto.

3.1.11 Bases de datos relacionales

Las bases de datos relacionales son sistemas que almacenan datos en tablas estructuradas con filas y columnas. Utilizan el lenguaje SQL para manipular y consultar datos, permitiendo relaciones entre tablas mediante claves primarias y foráneas. Son ideales para aplicaciones con datos estructurados y relaciones definidas, como sistemas de gestión de inventario, CRM y sistemas financieros. Ejemplos populares incluyen MySQL, PostgreSQL y Oracle Database.

3.1.11.1 PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado en aplicaciones web y empresariales. Ofrece una robusta arquitectura, características avanzadas y un alto nivel de seguridad. PostgreSQL es conocido por su capacidad para manejar grandes cantidades de datos y por su soporte para consultas complejas y transacciones. Como base de datos relacional, utiliza el lenguaje SQL para realizar consultas y manipular datos, lo que lo hace fácil de usar para desarrolladores y administradores de bases de datos.

3.1.12 Bases de datos no relacionales

Las bases de datos no relacionales son sistemas que almacenan y recuperan datos sin seguir un modelo de tabla tradicional. Son flexibles y escalables, lo que las hace ideales para manejar grandes volúmenes de datos no estructurados o semiestructurados, como documentos JSON, datos geoespaciales, y grafos. No requieren un esquema fijo, lo que permite una mayor agilidad en el desarrollo y la capacidad de adaptarse a cambios en los datos. Ejemplos populares incluyen MongoDB, Cassandra y Redis.

3.1.13 Aprendizaje en línea o E-learning

El E-Learning es una evolución de la formación que nace con la expansión de internet, que pone a disposición de los profesionales de la enseñanza virtual un nuevo entorno rico en recursos. A diferencia del aprendizaje a distancia, esta modalidad permite crear espacios virtuales para impartir clases, ejercicios interactivos, foros de discusión para ampliar los contenidos impartidos e incluso simulaciones reales para poner en práctica los conocimientos (Herrera, 2021).

El E-learning Ofrece flexibilidad en términos de tiempo y ubicación, permitiendo a los estudiantes acceder al contenido del curso en cualquier momento y desde cualquier lugar con conexión a internet. Esta modalidad de aprendizaje ha ganado popularidad debido a su accesibilidad y conveniencia, eliminando las barreras geográficas y temporales asociadas con la educación tradicional. Además, proporciona una amplia gama de recursos multimedia y herramientas interactivas que enriquecen la experiencia de aprendizaje. Ejemplos comunes de plataformas de aprendizaje en línea incluyen Moodle, Platzi y Udemy.

3.1.14 Aprendizaje en línea de tipo B-learning

El B-Learning, o aprendizaje semipresencial, emerge como una estrategia educativa híbrida que fusiona la riqueza de la interacción presencial con la flexibilidad del entorno virtual. Esta modalidad aprovecha las ventajas de ambos mundos, permitiendo la creación de experiencias de aprendizaje más dinámicas y adaptadas a las necesidades individuales. A través de la

combinación de clases presenciales tradicionales con plataformas en línea, se facilita el acceso a recursos digitales, actividades interactivas y espacios de colaboración virtual. El B-Learning no solo ofrece flexibilidad en cuanto a tiempo y ubicación, sino que también promueve la interacción directa entre estudiantes y profesores, fomentando un aprendizaje más profundo y significativo. Ejemplos de implementación del B-Learning incluyen la utilización de aulas virtuales complementarias a las clases presenciales, la realización de actividades en línea para reforzar los conceptos aprendidos en el aula, y la participación en foros de discusión para ampliar el debate y el intercambio de ideas.

3.1.14.1 Meiweb

Meiweb, acrónimo de Material Educativo Informático en la web, es una plataforma usada en la Universidad Industrial de Santander como una herramienta virtual de aprendizaje de tipo B-learning, diseñada para facilitar la interacción entre profesores y estudiantes. En esta plataforma se gestionan los contenidos de cursos relacionados con asignaturas como programación en la web y sistemas operacionales, ofreciendo recursos educativos y herramientas para llevar a cabo actividades evaluativas. Utilizado en la Universidad Industrial de Santander, Meiweb se destaca como un recurso integral que gestiona la asistencia, el material de curso, el calendario, las actividades y las evaluaciones programadas, promoviendo así un entorno educativo en línea efectivo y colaborativo.

3.1.14.2 Moodle

Moodle es una plataforma usada en la Universidad Industrial de Santander como una herramienta virtual de aprendizaje de tipo B-learning. Esta es de código abierto, lo que facilita la creación y administración de cursos en línea. Permite a los educadores crear contenidos interactivos, realizar seguimiento del progreso de los estudiantes y facilitar la comunicación en línea. Moodle es altamente personalizable y cuenta con una amplia comunidad de usuarios que contribuyen con plugins y recursos adicionales. Es una herramienta versátil y ampliamente utilizada en instituciones educativas de todo el mundo para ofrecer una experiencia de aprendizaje en línea efectiva y colaborativa.

4. Estado del arte

4.1 Antecedentes de la situación de estudio

En este apartado se abordan los antecedentes y trabajos relevantes del estado del arte relacionados con el e-learning, b-learning, las plataformas de aprendizaje virtual y los laboratorios clínicos que gestionan resultados mediante aplicaciones web.

En el ámbito educativo, destacan plataformas como:

- Moodle: Un LMS utilizado por diversas instituciones educativas en Colombia y en todo el mundo, que permite crear entornos de estudio personalizados y está disponible en varios idiomas.
- Platzi: Una plataforma enfocada principalmente en tecnología, ofreciendo cursos sobre programación, arquitectura de software y ciencia de datos, así como otros temas como finanzas e idiomas. Es popular entre estudiantes y profesionales que buscan mejorar sus habilidades y avanzar en sus carreras.
- Udeemy: Con más de 180,000 cursos disponibles y más de 44 millones de estudiantes, Udeemy se autodenomina como una tienda de aprendizaje virtual con el objetivo de mejorar las vidas de las personas a través del aprendizaje.

Además, se considerarán algunos laboratorios clínicos que realizan el proceso de muestra de resultados a través de aplicaciones web:

- Epic Systems: Con sede en Estados Unidos, Epic Systems es conocido por su sistema de registro médico electrónico (EMR) utilizado en muchos hospitales y sistemas de salud prominentes en todo el mundo.
- Higuera Escalante: Laboratorio clínico y banco de sangre altamente certificado, reconocido especialmente en el oriente colombiano. Además de sus servicios clínicos, lleva a cabo investigaciones para mejorar las prácticas y diagnósticos en los exámenes clínicos.

5. Metodología

El desarrollo del presente proyecto sigue una metodología basada en el marco de trabajo ágil Scrum, que permite una organización eficiente de las actividades y garantiza la flexibilidad necesaria para adaptarse a posibles cambios. Scrum es particularmente útil en proyectos donde los objetivos pueden evolucionar y donde se busca obtener resultados iterativos en plazos cortos.

El proyecto se estructura en tres etapas principales, cada una diseñada para abordar los objetivos específicos de manera organizada: planificación, desarrollo y revisión. Estas partes se detallan a continuación:

5.1 Planning

En esta fase inicial, se lleva a cabo la definición y organización del proyecto, sentando las bases para un desarrollo eficiente. Se incluyen las siguientes actividades:

- **Análisis del problema:** Investigar el ámbito del proyecto, en este caso, laboratorios clínicos, para comprender mejor las necesidades y las posibles soluciones.
- **Estudio de tecnologías:** Explorar herramientas y buenas prácticas relacionadas con las tecnologías seleccionadas (Angular y Node.js), con el objetivo de aplicarlas en el desarrollo.
- **Diseño del plan de trabajo:** Elaborar un diagrama de actividades y un backlog con las tareas organizadas en sprints, que incluyan metas específicas y factibles.
- **Definición de contenido académico:** Diseñar los materiales del curso "Programación en la web", integrando un enfoque práctico y la implementación de un prototipo funcional como recurso de aprendizaje.
- **Diseño de la guía básica de la metodología Scrum:** Explorar diferentes alternativas de contenido relacionado a Scrum y se diseña la mejor manera de presentar este contenido, con el fin de crear la guía básica de la metodología para el uso de los estudiantes.

5.2 Desarrollo

En esta etapa se ejecutan los sprints definidos en la planificación, siguiendo los principios de Scrum para iterar y mejorar continuamente el proyecto. Las actividades incluyen:

- **Creación de material teórico:** Generar documentos y recursos para que los estudiantes puedan comprender y aplicar los conceptos de las tecnologías utilizadas.
- **Diseño de ejercicios prácticos:** Proponer actividades que permitan consolidar los conocimientos adquiridos en clase.
- **Desarrollo del prototipo:** Construir una versión funcional del sistema de resultados de laboratorio clínico, asegurando su integración con los materiales educativos del curso.
- **Publicación de recursos:** Subir los contenidos del curso y el prototipo a la plataforma académica correspondiente para su uso por parte de los estudiantes.
- **Creación de la guía básica de la metodología Scrum:** Construir el archivo que contiene la información correspondiente a la guía básica de la metodología para los estudiantes y subir esta a las plataformas Moodle y Meiweb para su uso por parte de los estudiantes.

5.3 Revisión

La etapa final consiste en la evaluación y mejora del prototipo desarrollado, con el objetivo de garantizar un producto de alta calidad. Las actividades son:

- **Pruebas del prototipo:** Realizar pruebas funcionales y de rendimiento para identificar posibles errores y áreas de mejora.
- **Análisis de retroalimentación:** Recopilar comentarios de los usuarios (estudiantes y profesores) para orientar las mejoras.
- **Planificación de mejoras:** Establecer un plan para futuras iteraciones que permitan profesionalizar y optimizar la solución presentada.

6. Desarrollo de proyecto

6.1 Fase de planning

6.1.1 Herramientas

A continuación, se detalla el conjunto de herramientas empleadas en la construcción de la aplicación web clínico, así como los criterios que fundamentaron su selección.

6.1.1.1 Angular

Angular se posiciona actualmente Como uno de los framework frontend más populares y estables, ofreciendo una sólida base para el desarrollo de aplicaciones web. Su arquitectura modular, respaldada por Google, y su gran comunidad lo convierten en una opción robusta y confiable. Además, su arquitectura basada en componentes, facilita la creación de aplicaciones escalables y mantenibles.

6.1.1.2 Bootstrap

Al ser uno de los frameworks CSS más populares, Bootstrap cuenta con un ecosistema sólido de herramientas y plugins que amplían sus funcionalidades y hacen sencillo su uso, ahorrando tiempo y esfuerzo al no tener que construir componentes para interfaces de usuario desde cero. Permitiendo a los desarrolladores centrarse en otros aspectos de la construcción de la aplicación.

6.1.1.3 NodeJS y Express

Node.js y Express.js forman una combinación poderosa para el desarrollo backend moderno. Node.js, con su modelo de E/S no bloqueante y su vasto ecosistema de paquetes npm, proporciona un entorno de ejecución rápido y escalable. Express.js, por su parte, actúa como un framework minimalista y flexible que simplifica la creación de rutas, el manejo de solicitudes HTTP y la integración con otras tecnologías. Al utilizar esta combinación, los desarrolladores pueden construir aplicaciones web dinámicas, eficientes y escalables, aprovechando las ventajas de JavaScript tanto en el frontend como en el backend.

6.1.1.4 PostgreSQL

PostgreSQL destaca por su arquitectura robusta y su modelo de datos relacional extendido. Ofrece un conjunto completo de tipos de datos, incluyendo números, cadenas de texto, fechas, horas, y tipos de datos definidos por el usuario. Además, PostgreSQL proporciona un lenguaje de consulta poderoso (SQL) y herramientas avanzadas para la administración de bases de datos, como copias de seguridad, restauración y replicación.

6.1.2 Control de tareas

6.1.2.1 Trello

Para la gestión de tareas en el proyecto se usó Trello, la cual es una plataforma de software para la gestión de proyectos, principalmente utilizada para realizar el seguimiento de errores y gestionar tareas en proyectos de desarrollo de software. Además, Trello facilita el uso de metodologías ágiles como Scrum, lo que permite a los equipos adaptarse rápidamente a los cambios y entregar aportes y avances a un proyecto de una manera más eficiente.

6.1.3 Control de versiones

6.1.3.1 Git y Github

Git es un Sistema de Control de Versiones Distribuido utilizado para guardar diferentes versiones de un archivo (o un conjunto de archivos) para que cualquier versión sea recuperable cuando se desee, lo cual es de suma importancia a la hora de trabajar en desarrollo de software. GitHub por otra parte es una plataforma basada en la web donde los usuarios pueden alojar repositorios Git. Facilita compartir y colaborar fácilmente en proyectos con cualquier persona en cualquier momento (Castellanos, 2021).

6.1.4 Despliegue

Para el despliegue de la aplicación, se seleccionaron servicios de hosting compatibles con el stack tecnológico para garantizar disponibilidad continua y acceso remoto. Se utilizó Render para el backend, por su eficiencia en operaciones de servidor, y Netlify para el frontend,

asegurando un rendimiento óptimo y una experiencia de usuario rápida y confiable desde cualquier ubicación.

6.1.4.1 Render

Render es una plataforma en la nube diseñada para facilitar a desarrolladores y equipos el alojamiento y la exposición de aplicaciones web y servicios en la nube de manera sencilla, ya que proporciona toda la infraestructura necesaria. Esta herramienta permite integraciones fluidas con GitHub y GitLab, lo que agiliza el proceso de despliegue. Además, cuenta con la capacidad de escalar automáticamente los recursos, asegurando que el rendimiento de la aplicación no se vea afectado incluso ante aumentos en la demanda.

6.1.4.2 Netlify

Netlify es una plataforma en la nube que simplifica el despliegue y la gestión de aplicaciones web estáticas y dinámicas, ofreciendo a desarrolladores y equipos una solución integral para alojar sus proyectos de manera eficiente. Con integraciones nativas con GitHub y GitLab, Netlify agiliza el proceso de despliegue continuo, permitiendo que los cambios en el código se reflejen automáticamente. Por otro lado, su infraestructura global y escalable también garantiza un rendimiento óptimo gracias a su red de servidores de alta velocidad.

6.1.5 Pruebas

Para comprobar la calidad del aplicativo realizado y el correcto cumplimiento de los requerimientos establecidos, es necesario pasar por una fase de pruebas o también llamado testing. Al realizar pruebas exhaustivas, se identifican y corrigen errores, fallos y cualquier desviación de los requisitos establecidos. Esto no solo mejora la experiencia del usuario final, sino que también reduce los costos de mantenimiento a largo plazo.

- **Pruebas end-to-end:** El e2e testing es una metodología que comprueba el funcionamiento de toda una aplicación, de principio a fin, por ello, se le conoce como pruebas de extremo a extremo o end-to-end (e2e) (¿Qué es End-to-End Testing?, 2018).

- **Pruebas unitarias:** son el proceso en el que se prueba la unidad funcional de código más pequeña. Las pruebas de software ayudan a garantizar la calidad del código y son una parte integral del desarrollo de software. Una práctica recomendada en el desarrollo de software es escribir el software como unidades pequeñas y funcionales, y luego escribir una prueba unitaria para cada unidad de código. Puede escribir primero pruebas unitarias como código (Amazon, s.f).

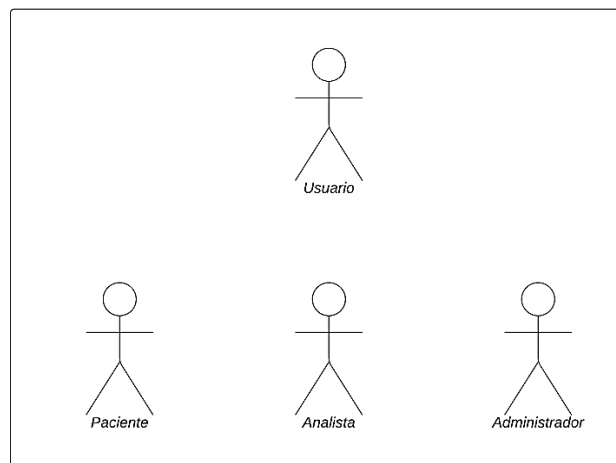
6.1.5.1 Cypress

Cypress es una herramienta de prueba de front-end de próxima generación construida para la web moderna. Abordamos los equipos clave de puntos de dolor cara al probar aplicaciones modernas y mantener suites de prueba. Cypress puede probar cualquier cosa que se ejecute en un navegador y obtener información de la superficie sobre cómo mejorar la salud de su conjunto de pruebas y la calidad de su aplicación (Cypress testing solutions, s.f).

6.1.6 Análisis de Desarrollo

En la siguiente sección se encuentran los requerimientos funcionales y no funcionales que hicieron parte de la construcción de la nueva versión para el aplicativo web clínico, adoptando los requerimientos de la primera versión y con algunas funcionalidades añadidas.

Figura 4. Actores en el sistema



6.1.6.1 Requerimientos funcionales

A continuación, se listan los requerimientos funcionales, basados en los requerimientos funcionales que se plantearon en la primera versión del aplicativo web clínico con algunas funcionalidades añadidas.

Tabla 1. Requerimiento funcional 1

Id: RF001	Nombre: Crear cuenta	Prioridad: Alta
Responsable	Analista, Paciente	
Descripción		
Registro de usuarios, los usuarios se registran en el sistema y este les asigna por defecto el rol 'Paciente'.		
Consideraciones	Se ingresan datos del usuario, tales como: documento, nombre, apellidos, correo, entre otros.	

Tabla 2. Requerimiento funcional 2

Id: RF002	Nombre: Ver información de examen a realizar	Prioridad: Alta
Responsable	Paciente	
Descripción		
Se deben poder visualizar los datos correspondientes a los exámenes disponibles, información como: nombre del examen, descripción		
Consideraciones	El paciente contará con información previa del examen que podrá visualizar antes de que sea realizado	

Tabla 3. Requerimiento funcional 3

Id: RF003	Nombre: Descarga de resultados	Prioridad: Alta
Responsable	Paciente, Analista	
Descripción		
<p>Previo a descargar el resultado se debe visualizar la información del resultado, nombre, fecha de subida, fecha de realización del examen. Adicional, el pdf debe tener concordancia con lo el resultado seleccionado a descargar</p>		
Consideraciones	<p>El analista podrá contar con el resultado para brindar su diagnóstico. El paciente por otro lado, tiene la oportunidad de descargarlo para saber de sus propios resultados</p>	

Tabla 4. Requerimiento funcional 4

Id: RF004	Nombre: Cargar resultado	Prioridad: Alta
Responsable	Analista, Administrador	
Descripción		
<p>El analista podrá subir los resultados al sistema, esta acción también la puede realizar el administrador del sistema.</p>		
Consideraciones	<p>Para cargar el resultado se deben llenar los siguientes campos: Carga del PDF, con el resultado correspondiente, nombre del usuario y examen correspondiente al que se sube dicho resultado.</p>	

Tabla 5. Requerimiento funcional 5

Id: RF005	Nombre: Recuperar contraseña	Prioridad: Media
Responsable	Paciente, Analista	
Descripción		
El paciente o analista puede recuperar su contraseña siguiendo el debido proceso para poder generar una nueva contraseña.		
Consideraciones	El paciente o analista deberá realizar el proceso de cambio de contraseña en el lapso establecido antes de que se venza el token de recuperación.	

Tabla 6. Requerimiento funcional 6

Id: RF006	Nombre: Actualizar contraseña	Prioridad: Media
Responsable	Paciente, Analista	
Descripción		
El paciente o analista podrá cambiar su contraseña cuando lo vea necesario sin embargo debe cumplir con los requisitos establecidos.		
Consideraciones	El cambio de contraseña está sujeto a las condiciones establecidas para la creación de una nueva contraseña como el uso de una mayúscula, una minúscula, número mínimo de caracteres y un símbolo.	

Tabla 7. Requerimiento funcional 7

Id: RF007	Nombre: Actualizar contraseña	Prioridad: Media
Responsable	Paciente, Analista	
Descripción		
El paciente o analista podrá cambiar su contraseña cuando lo vea necesario sin embargo debe cumplir con los requisitos establecidos.		
Consideraciones	El cambio de contraseña está sujeto a las condiciones establecidas para la creación de una nueva contraseña como el uso de una mayúscula, una minúscula, número mínimo de caracteres y un símbolo.	

6.1.6.2 Requerimientos no funcionales

Tabla 8. Requerimiento no funcional 1

Id: RNF001	Nombre: Protocolo de seguridad	Prioridad: Alta
Responsable	Equipo de desarrollo	
Descripción		
El paciente o analista podrá cambiar su contraseña cuando lo vea necesario sin embargo debe cumplir con los requisitos establecidos.		
Consideraciones	Al equipo de desarrollo se le concede la confiabilidad de la seguridad del aplicativo, por ende, se recomienda someter a este a pruebas de seguridad.	

Tabla 9. Requerimiento no funcional 2

Id: RNF002	Nombre: Diseño de Interfaz atractiva	Prioridad: Alta
Responsable	Equipo de desarrollo	
Descripción		
El sitio web debe poseer interfaces gráficas bien formadas y llevar en su interfaz los colores adecuados para un laboratorio clínico		
Consideraciones	Los colores pertenecientes a la interfaz deben ir relacionados con el campo al cual pertenece el aplicativo, es decir, el sector salud.	

Tabla 10. Requerimiento no funcional 3

Id: RNF00	Nombre: Disponibilidad tiempo completo	Prioridad: Alta
Responsable	Equipo de desarrollo	
Descripción		
El sistema debe tener una disponibilidad del 99,99% de las veces en que un usuario intente acceder.		
Consideraciones	El software debe permanecer en la web todo el tiempo.	

Tabla 11. Requerimiento no funcional 4

Id: RNF004	Nombre: Soporte de usuarios	Prioridad: Alta
Responsable	Software.	
Descripción		
El sistema debe ser capaz de operar adecuadamente con hasta 10.000 usuarios con sesiones concurrentes.		
Consideraciones	El software debe responder adecuadamente a cada una de las peticiones que realice cada usuario.	

Tabla 12. Requerimiento no funcional 5

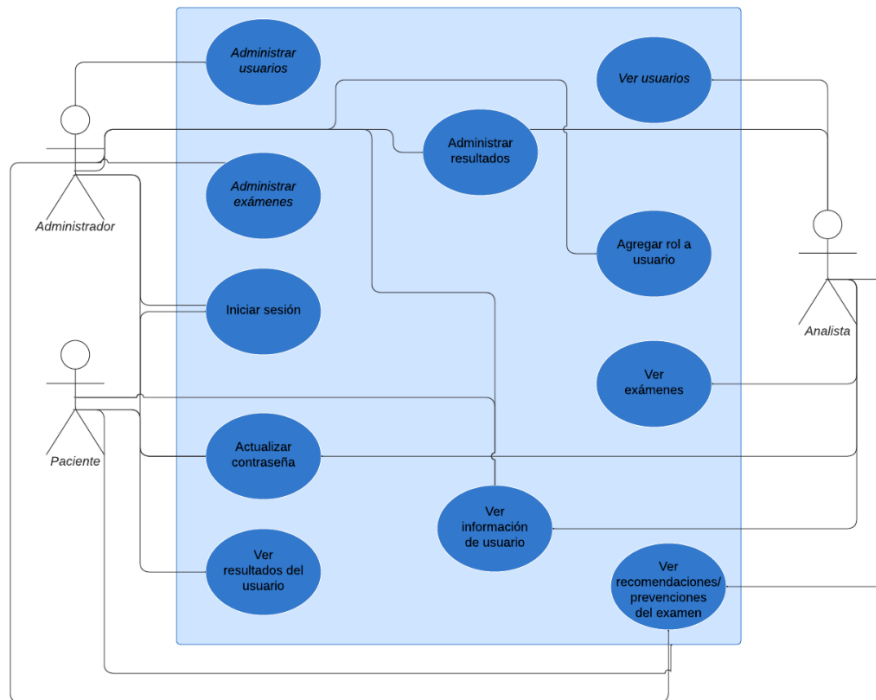
Id: RNF005	Nombre: Tiempo de respuesta	Prioridad: Media
Responsable	Software.	
Descripción		
Toda funcionalidad del sistema y transacción del software debe responder al usuario en menos de 5 segundos.		
Consideraciones	El aplicativo deberá tener un tiempo de respuesta inferior a los 5 segundos máximo por petición realizada.	

Tabla 13. Requerimiento no funcional 6

Id: RNF006	Nombre: Control de errores	Prioridad: Media
Responsable	Equipo de desarrollo, Software.	
Descripción		
El software debe proporcionar mensajes de error que sean informativos y orientados a usuario final.		
Consideraciones	Si se presenta algún tipo de error que requiera el ajuste por parte de los desarrolladores deberá ser solucionado en un tiempo menor a un día hábil.	

6.1.6.3 Casos de uso

Figura 5. Diagrama de casos de uso



6.1.6.4 Diagrama de actividades

Figura 6. Diagrama de inicio de sesión

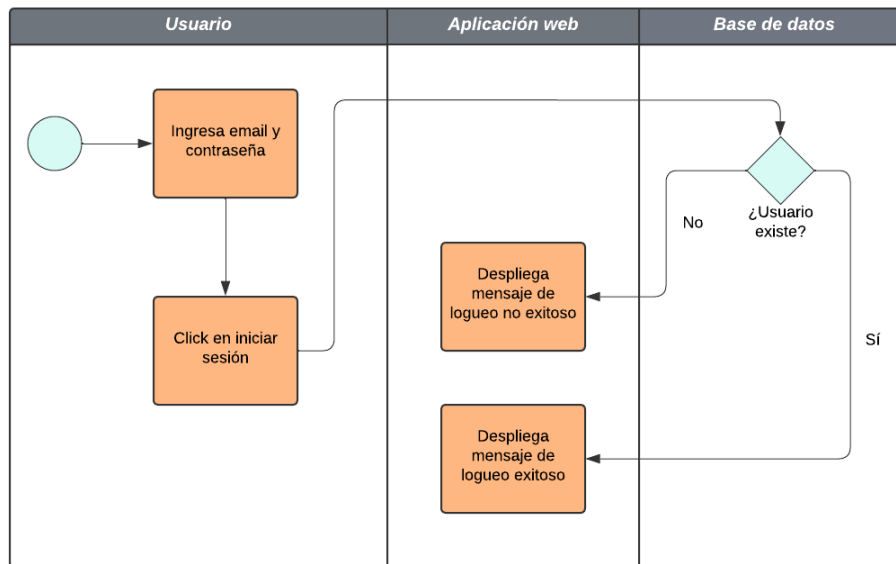


Figura 7. Diagrama para registro creación de cuenta

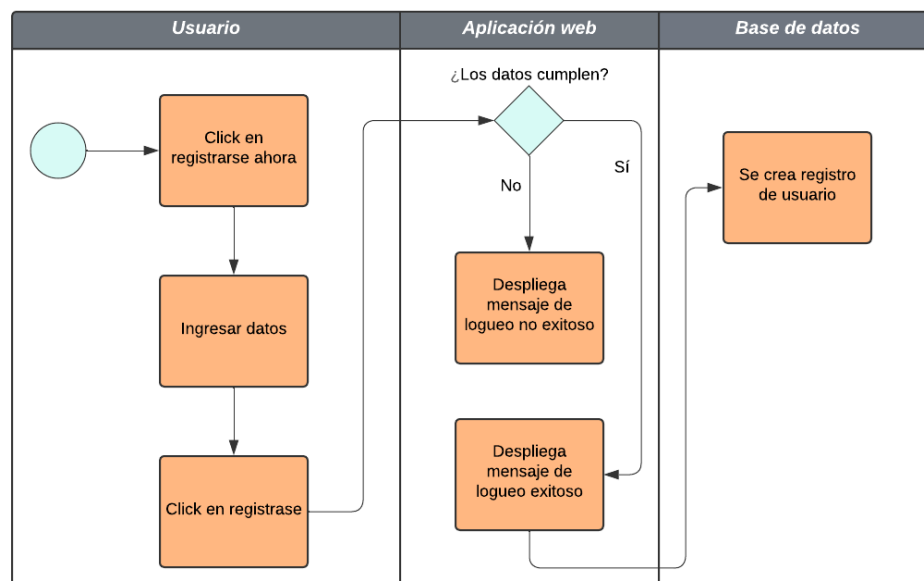


Figura 8. Diagrama para ver resultados

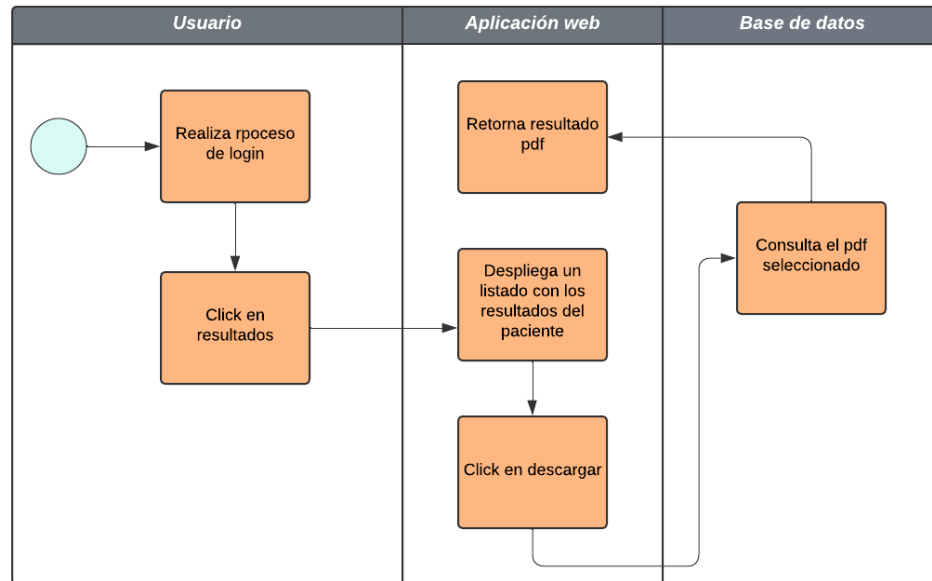


Figura 9. Diagrama para subir resultados

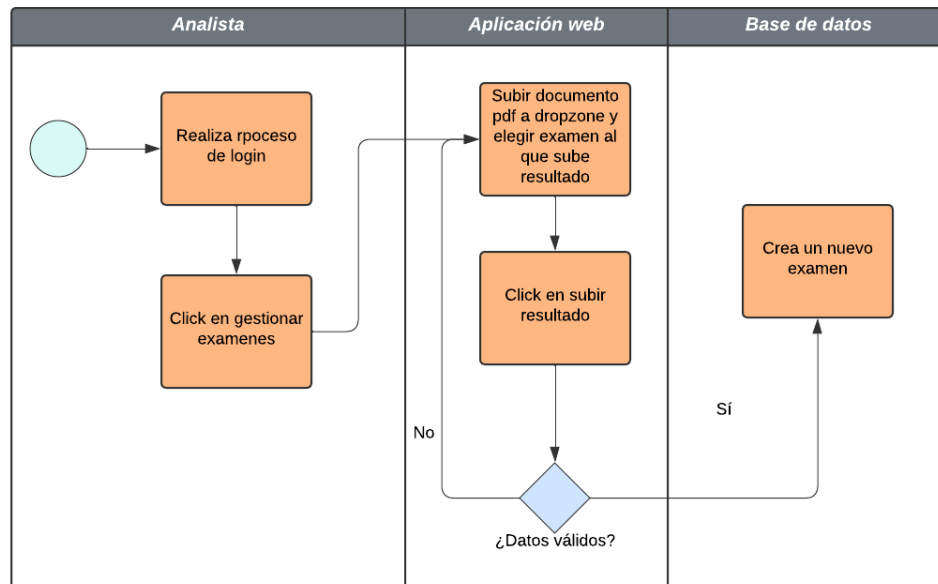
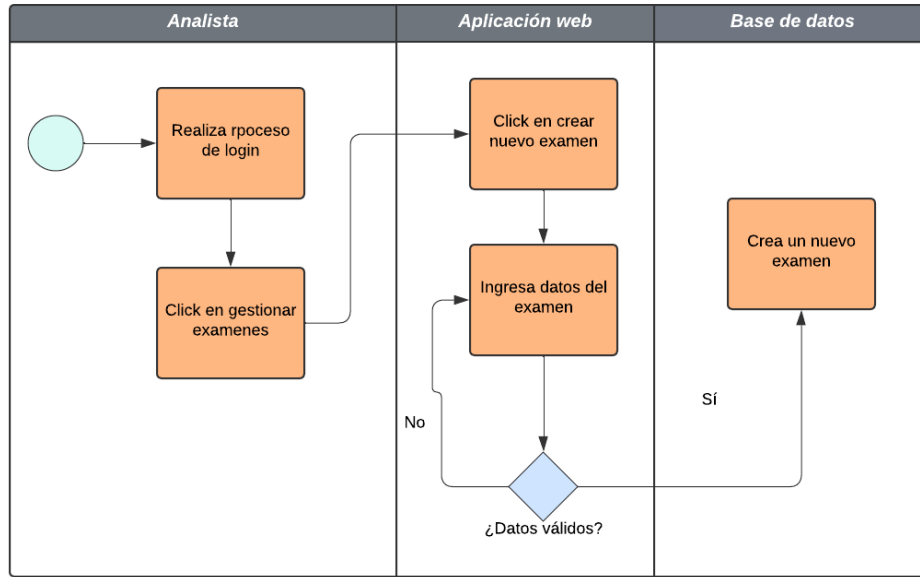


Figura 10. Diagrama para crear un examen

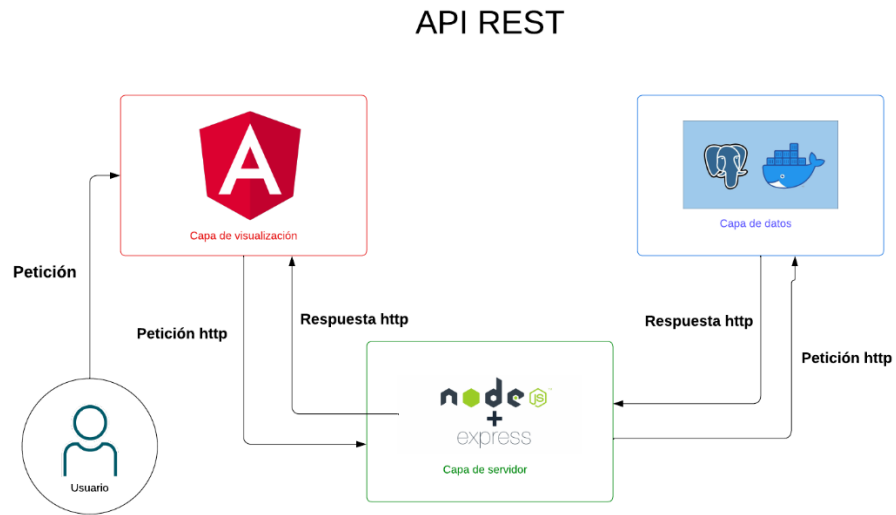


6.1.6.5 Arquitectura

La arquitectura del sistema se basa en un modelo de tres capas: presentación, lógica de negocio y datos. La capa de presentación, desarrollada con Angular, se comunica con la capa de lógica de negocio a través de una API REST construida con Node.js y Express.js. La capa de lógica de negocio, a su vez, interactúa con una base de datos PostgreSQL para almacenar y recuperar los datos.

La arquitectura en capas es un patrón de arquitectura software usada en la gran mayoría de sistemas. Se centra en una distribución jerárquica de las roles y responsabilidades proporcionando una separación efectiva de las preocupaciones. Al pensar en un sistema en términos de capas, se imaginan los principales subsistemas de software ubicados de la misma forma que las capas de un pastel, donde cada capa descansa sobre la inferior.

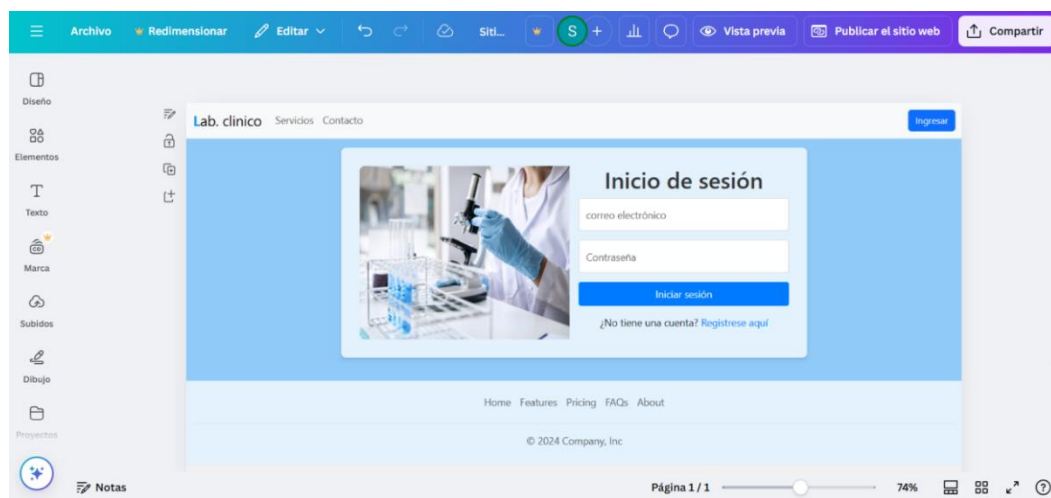
Figura 11. Arquitectura rest



6.1.7 Interfaz de usuario

El diseño de la interfaz de usuario se basó en un enfoque centrado en el usuario y se llevó a cabo utilizando Canva. A través de mockups o plantillas que se siguieron con fidelidad, visualizamos la estructura y el flujo de la aplicación. Aplicamos los principios de diseño para garantizar una experiencia visual coherente con los requerimientos planteados y atractiva a la vista del usuario.

Figura 12. Mockup inicio de sesión

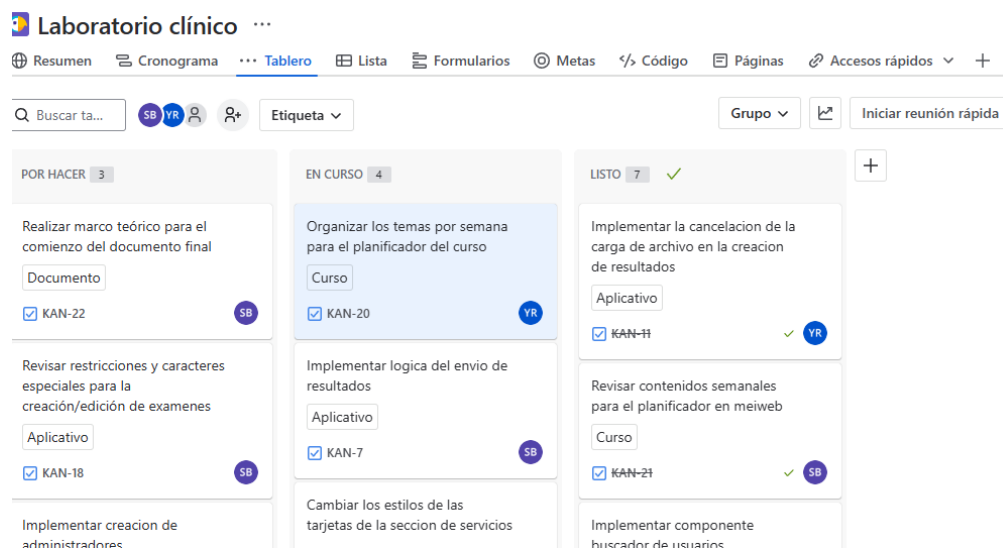


Para garantizar una experiencia de usuario coherente y memorable, se estableció una paleta de colores específica basada en los perfiles de nuestros usuarios. Asimismo, los flujos de usuario fueron diseñados de manera intuitiva para guiar al usuario de forma clara y eficiente.

6.2 Fase de desarrollo

A continuación, se describen las tres partes de desarrollo que tuvieron lugar en el proyecto, una para para la construcción del aplicativo web clínico y otra para la construcción del modelo de curso.

Figura 13. Tablero de tareas



6.2.1 Aplicativo web clínico

El aplicativo web clínico se estructura en dos partes, la parte backend y la parte frontend, cada una de estas partes con su respectivo repositorio alojado en Github. Para el control del desarrollo se usó el marco de desarrollo ágil Scrum y Trello para la gestión de tareas. Para obtener información más detallada sobre cómo utilizar el aplicativo, se recomienda consultar el Apéndice C - Manual de usuario para el uso del Aplicativo Clínico.

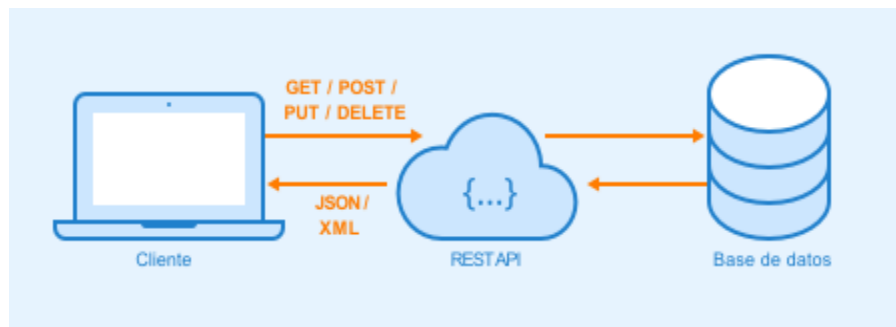
Figura 14. Lista de tareas aplicativo clínico

<input type="checkbox"/>	Tipo	Clave	Resumen	Estado	Persona asignada	Fecha d
<input type="checkbox"/>	<input checked="" type="checkbox"/>	KAN-1	Poner en funcionamiento la base de datos	FINALIZADA	YR Yezith Rincón	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	KAN-5	Implementar feature de envio de correo electrónico	FINALIZADA	SB Santiago Ariza Briceño	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	KAN-9	Refactorizar middlewares	FINALIZADA	YR Yezith Rincón	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	KAN-2	Refactorizar los servicios	FINALIZADA	SB Santiago Ariza Briceño	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	KAN-11	Implementar la cancelacion de la carga de archivo en la crea...	FINALIZADA	YR Yezith Rincón	
<input type="checkbox"/>	<input checked="" type="checkbox"/>	KAN-7	Implementar logica del envio de resultados	EN CURSO	SB Santiago Ariza Briceño	

6.2.1.1 Desarrollo backend

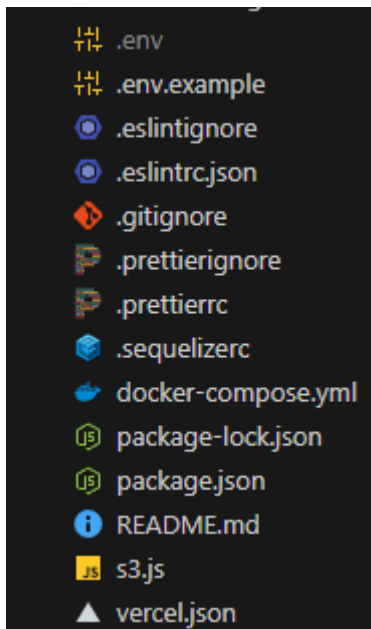
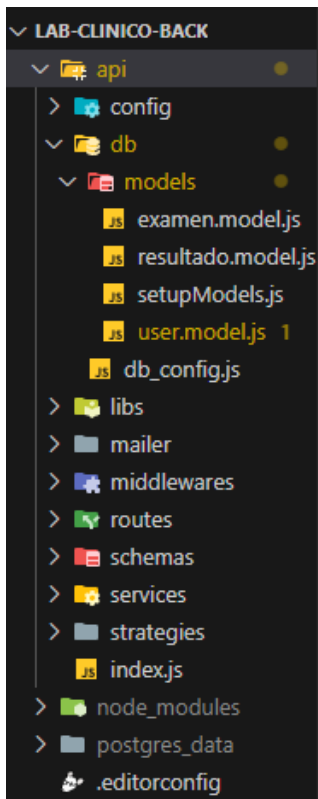
Para el desarrollo del backend de la aplicación se siguió la arquitectura ApiREST. Se usó también el stack de tecnologías PEAN, PostgreSQL para la gestión de la base de datos junto a Sequelize como ORM, Express y NodeJS para la capa de servidor.

Figura 15. Arquitectura api-rest



Adaptado de seobility.com

Figura 16. Estructura de carpetas backend



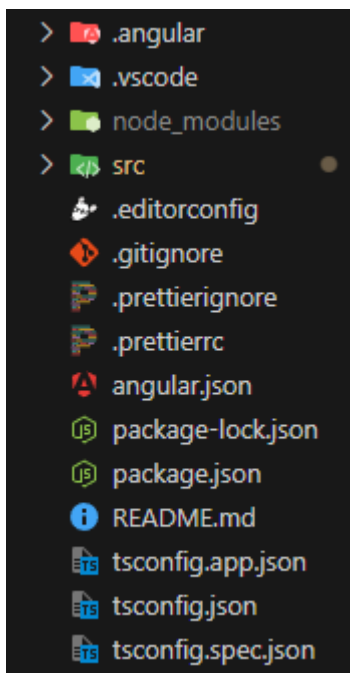
- **config**: es la carpeta que tiene el archivo que llama las variables de entorno.
- **db**: contiene todo lo relacionado a la base de datos y/o migraciones.

- `models`: contiene los modelos de los datos que representan la lógica de la base de datos.
- `libs`: contiene los archivos para hacer posible la conexión con la capa de datos.
- `mailer`: contiene los archivos necesarios para el envío de correos automáticos por parte de la aplicación.
- `middlewares`: contiene los archivos de los middlewares de verificación, autenticación y control de errores.
- `routes`: en esta carpeta se definen las rutas del aplicativo junto con `express`. Por cada entidad existe un `entidad.router` que representa las rutas para esta.
- `schemas`: se encuentran los distintos esquemas que validan datos de entrada para que cumplan con los requerimientos planteados.
- `services`: se guardan los archivos para los servicios, que son los encargados de manejar las peticiones `http` para cada `endpoint` del backend.
- `strategies`: en esta carpeta se encuentran distintos archivos que corresponden a las estrategias disponibles para autorización y autenticación.
- `node_modules`: almacena las librerías y módulos externos que utiliza el proyecto. Se crea automáticamente.
- `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.
- `.eslintignore`: archivo que especifica a `eslint` los archivos y directorios a omitir.
- `.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.
- `package-lock.json`: almacena las versiones de las dependencias instaladas.
- `package.json`: configuración principal del proyecto, contiene información detallada de todas las dependencias instaladas.
- `README.md`: archivo que contiene información y documentación acerca del proyecto.
- `s3.js`: archivo que se encarga de interactuar con el servicio de almacenamiento en la nube de Amazon S3, específicamente desde una aplicación Node.js.

6.2.1.2 Desarrollo frontend

Figura 17. Estructura de carpetas frontend

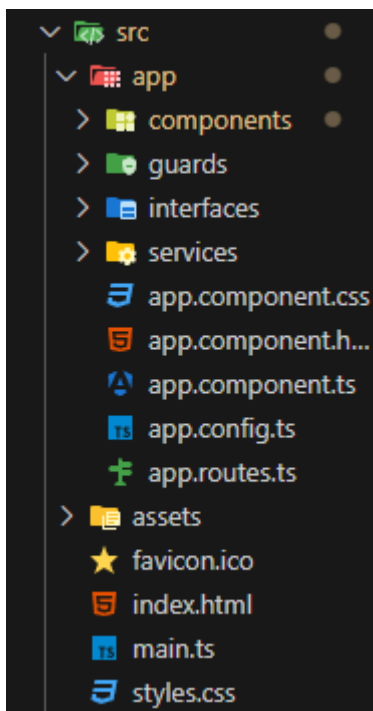


- `Editorconfig`: archivo que establece y mantiene la consistencia en el estilo de

codificación del proyecto.

- `.gitignore`: define los archivos y directorios que no deben incluirse al realizar commits en Git.
- `angular.json`: almacena información sobre la estructura del proyecto, las dependencias, las opciones de compilación y otras configuraciones esenciales para el desarrollo y construcción del proyecto.
- `Karma.conf.js`: configura y personaliza el entorno necesario para ejecutar pruebas unitarias.
- `Package.json`: Se encarga de administrar dependencias, scripts, metadatos y otras configuraciones del proyecto.
- `Tsconfig.json`: archivo de configuración de TypeScript que define las opciones y ajustes para la compilación con el compilador TypeScript (`tsc`).

Figura 18. Estructura de carpetas frontend carpeta src



En la carpeta src se encuentra todo el contenido de archivos donde se almacena el código fuente de la aplicación. Dichos archivos están separados según su categoría que puede ser: componentes (components), guardianes (guards), interfaces, servicios (services) y elementos visuales (assets).

Figura 19. Vista inicio de sesión



Figura 20. Vista registrarse



Figura 21. Vista inicial home



Figura 22. Vista de servicios



Figura 23. Vista resultados por paciente

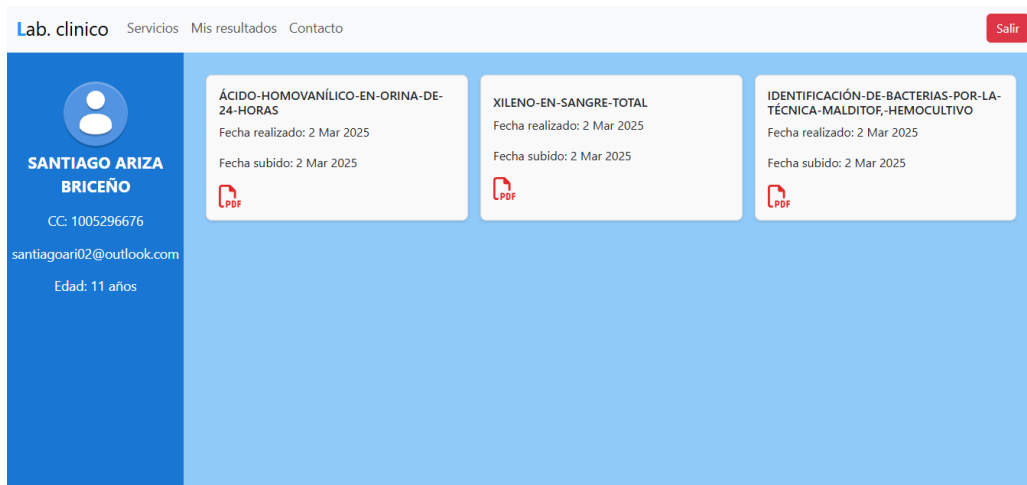


Figura 24. Vista subir resultados

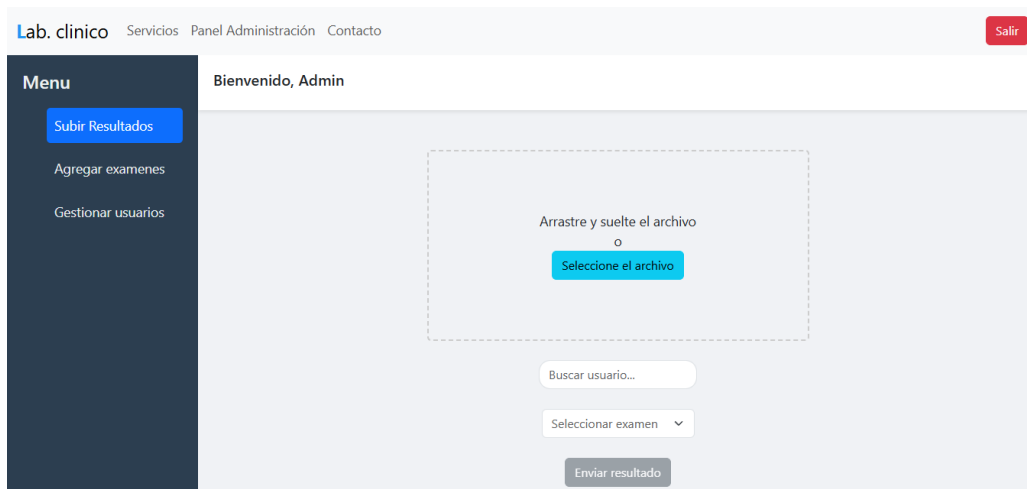


Figura 25. Vista crud exámenes

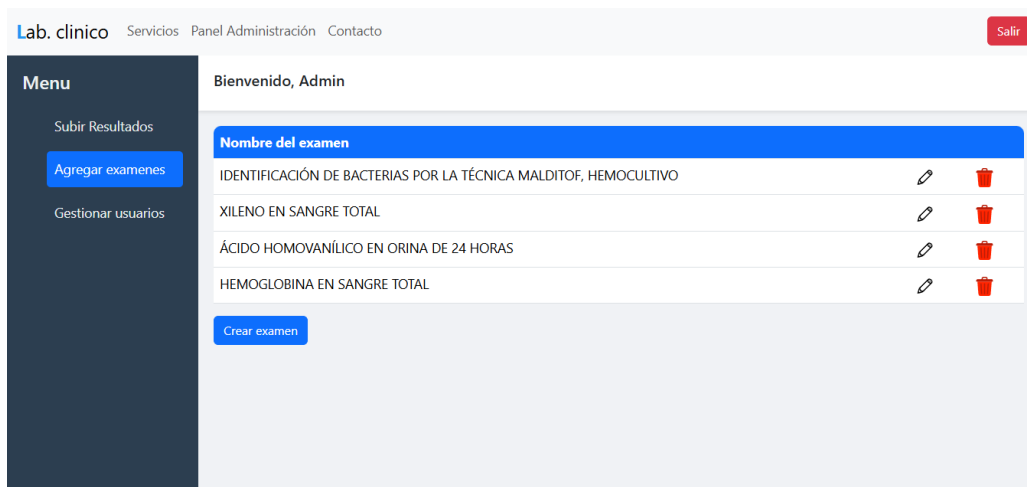


Figura 26. Vista modal para creación y edición de exámenes

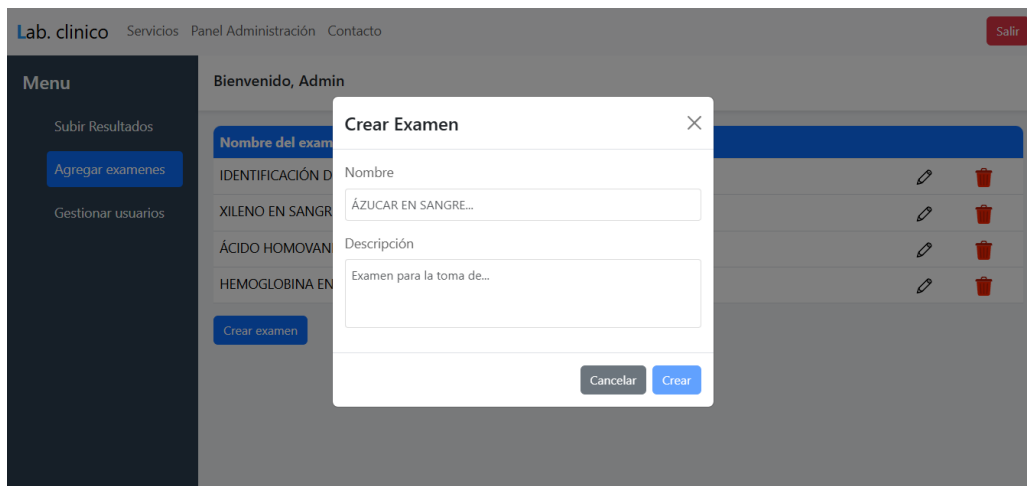


Figura 27. Vista gestión de usuarios

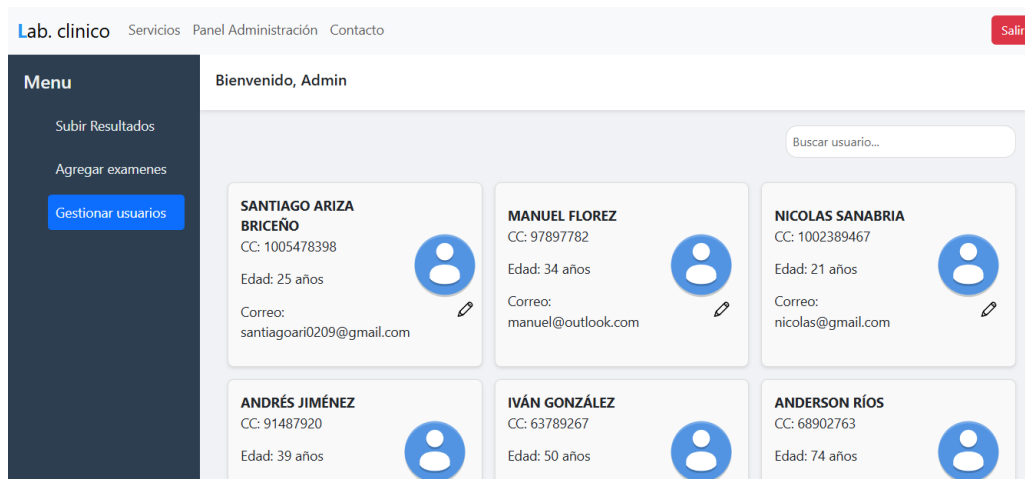


Figura 28. Vista modal para la gestión/edición de información de usuarios

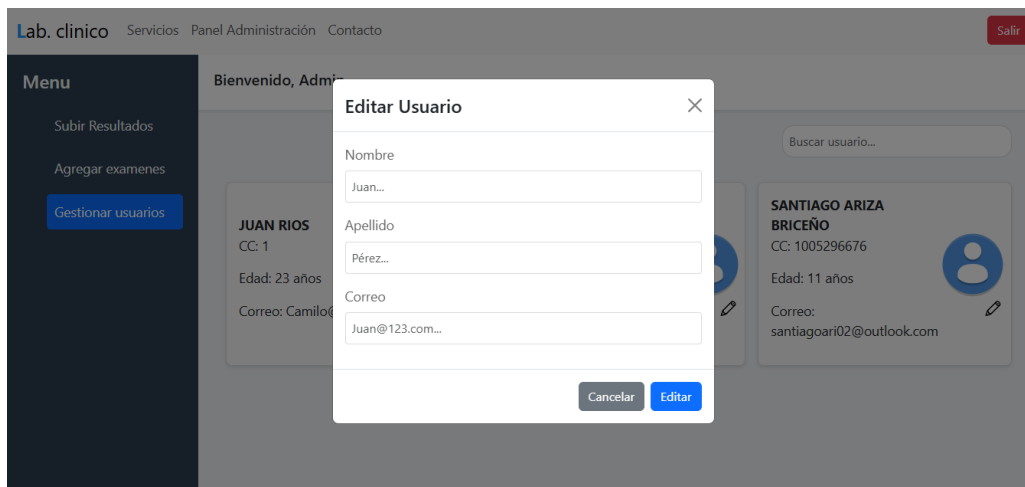


Figura 29. Vista página de contacto

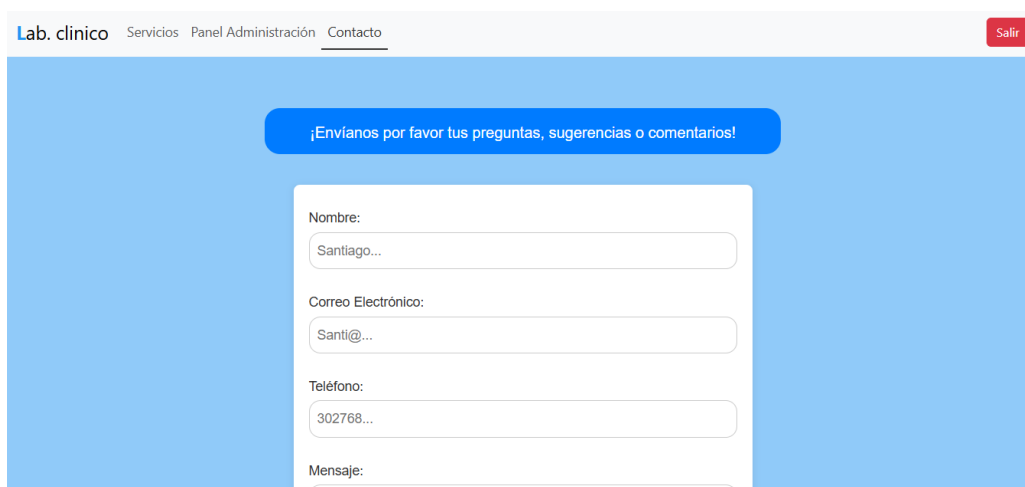


Figura 30. Vista página no encontrada o 404



6.2.2 Curso Programación en la Web

A continuación, se describen los procesos que se realizaron para el desarrollo del modelo de curso, el cual fue creado con el fin de implementar los conocimientos que fueron adquiridos durante el proceso de desarrollo del aplicativo web clínico. Para obtener información más detallada sobre cómo utilizar el curso, se recomienda consultar el Apéndice A - Manual de usuario para el uso del curso virtual Programación en la Web en la plataforma Moodle y el Apéndice B - Manual de usuario para el uso del curso virtual Programación en la Web en la plataforma Meiweb.

El curso ha sido diseñado para ofrecer una formación completa en desarrollo web. Desde los fundamentos de HTML, CSS y JavaScript, hasta tecnologías más avanzadas como Angular en conjunto con Typescript, Node.js y Express, el contenido ha sido estructurado de manera progresiva para facilitar el proceso de aprendizaje. También, Se ha puesto especial énfasis en la práctica, incluyendo numerosos ejercicios y proyectos para que los estudiantes apliquen los conocimientos adquiridos durante la programación semanal que se estructuró tomando en cuenta el calendario del semestre universitario.

Figura 31. Lista de tareas para creación del curso

Clave	Resumen	Estado	Persona asignada	Fecha de vencimiento	Etiquetas
KAN-21	Revisar contenidos semanales para el planificador en meiw...	FINALIZADA	SB Santiago Ariza Briceño		Curso
KAN-20	Organizar los temas por semana para el planificador del curso	FINALIZADA	YR Yezith Rincón		Curso
KAN-23	Revisar y filtrar los temas que irán incluidos en la guía básica ...	FINALIZADA	SB Santiago Ariza Briceño		Curso
KAN-24	Analizar cuáles son los mejores tipos de actividades para los ...	FINALIZADA	YR Yezith Rincón		Curso
KAN-25	Establecer un documento en donde se guarden los contenid...	FINALIZADA	SB Santiago Ariza Briceño		Curso
KAN-26	Actualizar el contenido del curso en la plataforma de Meiw...	FINALIZADA	SB Santiago Ariza Briceño		Curso

A la hora de realizar el planificador del curso tanto en el Moodle como en Meiw... se tuvieron en cuenta dos tipos de competencias para fomentar el aprendizaje de los estudiantes; las competencias prácticas y las competencias teóricas, las cuales están definidas para cada una de las semanas en el planificador y permiten tener una guía clara de lo que se busca lograr aprender en cada una de estas.

Figura 32. Planificador en Moodle del curso Programación en la Web con Angular

Introducción Objetivos y Políticas Cronograma de Evaluaciones **Planificador** Materiales y recursos Extras

Planificador **Semana 1** Semana 2 Semana 3 Semana 4 Semana 5 Semana 6 Semana 7 Semana 8 Semana 9

Semana 10 Semana 11 Semana 12 Semana 13 Semana 14 Avances del proyecto final

Presentación y calificación del proyecto final

📁 **Semana 1**

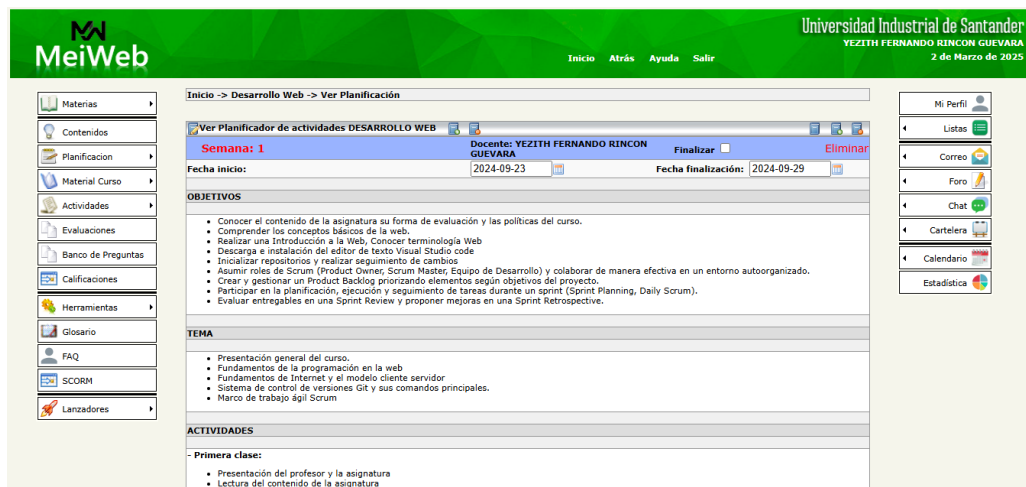
Competencias teóricas

- Presentación general del curso.
- Fundamentos de la programación en la web
- Fundamentos de Internet y el modelo cliente servidor
- Sistema de control de versiones Git y sus comandos principales.
- Marco de trabajo ágil Scrum

Competencias prácticas

- Conocer el contenido de la asignatura su forma de evaluación y las políticas del curso.
- Comprender los conceptos básicos de la web.
- Realizar una Introducción a la Web, Conocer terminología Web
- Descarga e instalación del editor de texto Visual Studio code
- Inicializar repositorios y realizar seguimiento de cambios
- Asumir roles de Scrum (Product Owner, Scrum Master, Equipo de Desarrollo) y colaborar de manera efectiva en un entorno autoorganizado.
- Crear y gestionar un Product Backlog priorizando elementos según objetivos del proyecto.
- Participar en la planificación, ejecución y seguimiento de tareas durante un sprint (Sprint Planning, Daily Scrum).
- Evaluar entregables en una Sprint Review y proponer mejoras en una Sprint Retrospective.

Figura 33. Planificador en Meiweb del curso Programación en la Web con Angular



6.2.2.1 Contenido del curso

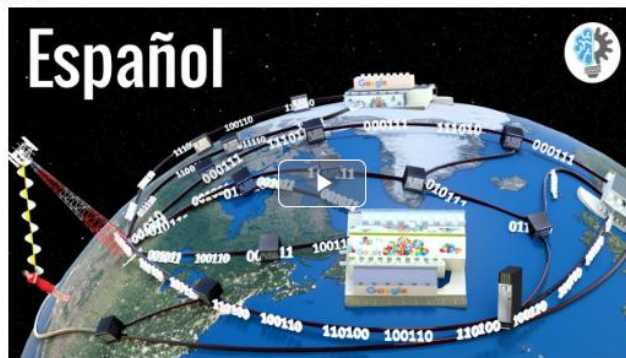
El contenido del curso se encuentra dividido en semanas, cada semana cuenta con enlaces a artículos, vídeos explicativos, vídeos tutoriales, proyectos para desarrollar ejercicios. Esto para la mejora en la experiencia de aprendizaje de los estudiantes.

Figura 34. Contenido del curso semana 1 Programación en la Web con Angular

☐ Semana 1

- Primera clase:

- [Lectura cómo funciona la web](#)
- [Vídeo - ¿Cómo funciona la web?:](#)



- [Qué es el modelo cliente – servidor](#)
- [Vídeo explicación modelo cliente – servidor:](#)



6.2.2.2 Planificación del curso

Para fomentar un aprendizaje activo y colaborativo, así como para ayudar al profesor, se incorporaron actividades, ejercicios y tareas durante cada una de las semanas en el planificador, que permiten a los estudiantes trabajar en proyectos de forma conjunta y creativa. Estas actividades proporcionan una guía al profesor al momento de saber cómo estructurar la clase, por otro lado, brindan un espacio dinámico donde los estudiantes pueden compartir ideas, recursos y desarrollar soluciones innovadoras. Esto para generar un ambiente de aprendizaje más atractivo e interesante.

Figura 35. Ejemplo Actividades semana 2 en planificador

Actividades

- Primera clase:

- o Elaborar una página que **(Ver ejemplo S2_E1)**:
 1. Contenga las etiquetas básicas con un párrafo que presente un saludo a los compañeros y al profesor
 2. A la página anterior anexar cuatro etiquetas; una para la codificación, otra para responsive design y otras dos a su preferencia.
 3. A la página anterior colocar un favicon, es decir, una imagen, en la pestaña del navegador.
 4. Añadir una etiqueta para implementar un panel de bienvenida.
 5. Esta página llama a una segunda página que presenta una imagen flotante del profesor indicando su profesión, título, oficina y un saludo.
 6. Por último, esta página que contiene la información del profesor llama a una tercera página en donde se presente el contenido del curso (en forma resumida solo títulos).

- Segunda clase:

- o Elaborar una página colocando estilo a un párrafo. **(Ver ejemplo S2_E2)**.
 1. Añadir una imagen a la página y centrar la imagen, use tecnología flex de css.
 2. Crear un favicon (icono en la pestaña de la página).
 3. Colocar un estilo a todos los párrafos, introducción a los css selectores de elementos; name, Id, class PseuClases y PseudoElementos.
 4. Colocar anclas (anchor o marcas) a cada párrafo, inicio, mitad y fin luego navegar entre estos.
- o Elaborar una página que contenga el arco Iris aplicando colores y estilos a la frase "Bienvenidos a la Universidad". **(Ver ejemplo S2_E3)**.
 1. Crear una hoja de estilos llamada arcoiris.css y vincularla a la página anterior mediante la etiqueta link.
 2. Opcional: Estilos externos: Visualización con estilos externos sin tener que escribir una sola línea de css a una página incluir estilos desde el core de CSS:
- o Elaborar una página que contenga seis párrafos con frases célebres o párrafos de una canción, aplicar un estilo a la primera letra de comienzo de párrafo. **(Ver ejemplo S2_E4)**.
 1. Colocar anclas para navegar entre párrafo y párrafo, ir al comienzo y al final.
 2. La página debe contener etiquetas para facilitar el responsive design y una que permita indexar la página en los buscadores de la web.

6.2.3 Guía básica de la metodología Scrum

Para el desarrollo de la guía básica de la metodología Scrum, se llevaron a cabo dos partes, la parte de diseño en donde se analizaron los contenidos y materiales relacionados con Scrum disponibles en internet. Luego, en la parte de creación, se escogieron los contenidos y materiales que fueran más acordes con los objetivos de aprendizaje del curso y por supuesto, se llevó a cabo la creación del documento de la guía junto con otro archivo complementario para

el desarrollo del plan de proyecto final de la asignatura con Scrum. Además, se socializan estos documentos en una clase presencial de la asignatura de Programación en la Web en la Universidad Industrial de Santander.

Figura 36. Primera página de la Guía básica de la metodología Scrum

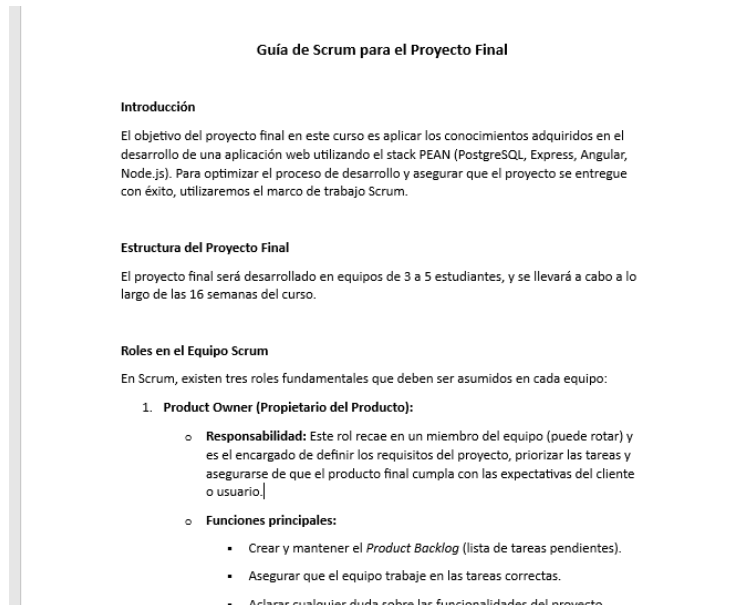
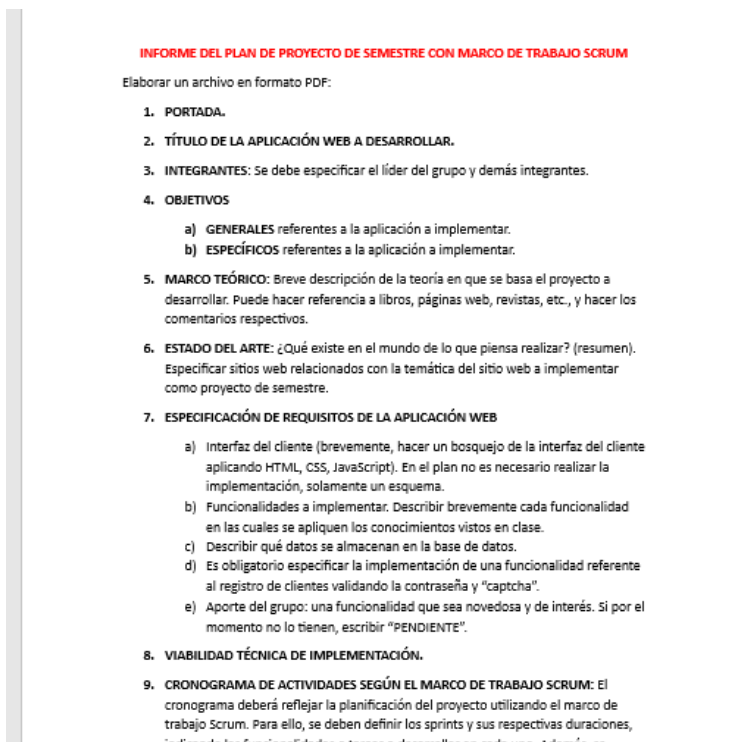


Figura 37. Primera página del formato de plan de proyecto con Scrum



6.3 Fase de evaluación

Esta sección está dividida en tres partes, la primera correspondiente a la parte de evaluación para el aplicativo web clínico, la segunda para la parte de evaluación del modelo de curso y la tercera para la parte de evaluación de la guía básica de la metodología Scrum .

6.3.1 Aplicativo web clínico

Con el fin de tener una evaluación del funcionamiento esperado del aplicativo, se planteó un plan de pruebas para evaluar las principales funcionalidades de este. Esto para comprobar que el aplicativo sea funcional, responda bien ante las interacciones del usuario y cumpla con los requerimientos que se plantearon para su desarrollo.

6.3.1.1 Plan de pruebas

Tabla 14. Plan de pruebas

Identificador	Objetivo	Entrada	Salida esperada	Salida	Comentario
RCU01	Registrar cuenta de usuario desde la página Home	Datos correctos para el registro del nuevo usuario	Mensaje de confirmación, notificando la creación de cuenta	Mensaje de éxito y usuario registrado	Ok, el sistema muestra un mensaje notificando la creación de la cuenta
RCU02	Registrar cuenta de usuario desde la página Home	Datos incorrectos para el registro del nuevo usuario	Mensaje de error	Mensaje de error	El sistema funciona correctamente, ya que los datos de registro son incorrectos

IS01	Iniciar sesión	Correo electrónico y contraseña correctas del usuario	Ingresa a la página principal según el rol de usuario logueado	Se dirige a la página principal del rol según lo esperado	Ok, funciona correctamente
IS02	Iniciar sesión datos incorrectos	Ingresa el correo o contraseña de manera incorrecta o no existen dichos datos en el sistema	Muestra mensaje de error señalando el campo que está mal ingresado	Falla en el inicio de sesión y vuelta a iniciar sesión	Ok, se muestra un mensaje notificando el campo que contiene errores
RC01	Recuperar contraseña	Click en recuperar contraseña, ingresa su correo registrado, le da click en enviar	Mensaje de éxito, diciendo que se ha enviado un correo	Mensaje de éxito. Se envía correo al email registrado	El correo se envía con link para ingresar a una página para cambiar la contraseña
RC02	Recuperar contraseña ,correo no existente	Click en recuperar contraseña, ingresa un correo no existente	Mensaje de error, diciendo que el correo no existe	Mensaje de error correspondiente al no encontrar el correo	Ok, mensaje notificando que no se encontró el correo
VR01	Ver resultados del usuario	Se inicia sesión de manera correcta	Se muestran los resultados correspondientes al usuario logueado	Se visualizan los resultados del usuario paciente	Ok

DR01	Descargar resultado (Paciente)	El paciente visualiza sus resultados y da click en descargar alguno de ellos	Resultado en pdf con la correcta información de usuario y el examen correspondiente	Se visualiza el resultado con los datos acordes a lo esperado por el paciente	Ok
SR01	Subir resultado (Analista)	El analista visualiza la vista subir resultados y sube correctamente el pdf y selecciona el examen correspondiente	Mensaje de éxito, notificando que el examen fue subido de manera correcta	Se ve el mensaje de éxito, advirtiendo que todo ha salido bien en la carga del examen	Ok
SR02	Subir resultado con formato incorrecto (Analista)	El analista visualiza la vista subir resultados y sube un formato distinto a pdf	Mensaje de error, diciendo que el archivo que desea subir, debería ser un pdf	Mensaje de error correspondiente a subir formato no permitido	Ok, se valida correctamente el formato de archivo
CE01	Crear/Editar examen (Administrador)	Ingresa a vista gestionar exámenes, da click en botón crear/editar examen, en el modal y llena datos correctamente	Mensaje creación/editación de examen exitosa	Se visualiza el mensaje de manera correcta	Ok

CE02	Crear/Editar examen con datos incompletos o erróneos (Administrador)	Ingresa a la vista gestionar exámenes y da click en crear/editar examen, en el modal llena los datos correctamente	No se habilita el botón para crear/editar examen y no es posible ejecutar la petición	Se evidencia que el botón de crear/editar no se activa al no llenar los campos correctamente	Ok
BE01	Borrar examen (Administrador)	Ingresa a la vista gestionar exámenes y selecciona el botón para eliminar un examen	Aparece un recuadro de confirmación para eliminar el examen, al confirmar aparece mensaje de eliminación exitosa	El mensaje de eliminación exitosa se muestra correctamente luego de que el administrador confirme la acción	Ok
GU01	Editar información de un usuario (Administrador)	Ingresa a la vista de gestión de usuarios y selecciona el usuario a editar, llena el modal correctamente	Aparece un mensaje de confirmación que indica la edición exitosa del usuario	De manera exitosa se confirma la edición de usuario	Ok
GU02	Editar información de un usuario de manera incorrecta (Administrador)	Ingresa a vista gestión de usuarios, selecciona usuario a editar, llena con datos incorrectos.	Aparece un mensaje anunciando que no fue posible la edición, no se realiza la petición	No se realiza la petición, dado el incorrecto llenado de los datos del usuario	Ok

6.3.2 Curso Programación en la Web

La nueva versión del modelo de curso de Programación en la Web con angular se estructuró de manera exitosa, actualizando contenidos y añadiendo nuevas secciones y actividades para cada uno de los contenidos que se estudian a lo largo de la asignatura. Así como también se planificaron de manera exitosa las nuevas actividades, ejercicios y tareas a los estudiantes, garantizando el aprendizaje de los principales temas relacionados al desarrollo web y con esto, ayudándoles al proceso de desarrollo de sus proyectos finales semestrales en la asignatura.

En la revisión del curso de programación web de la Universidad Industrial de Santander, el profesor evaluó positivamente la estructura del plan de estudios, asegurando que los contenidos se ajustaran al calendario semestral y que el material didáctico estuviera relacionado con el desarrollo del aplicativo, cumpliendo así con los objetivos de aprendizaje establecidos.

6.3.3 Guía básica de la metodología Scrum

La guía básica para la metodología Scrum se realizó con éxito, se organizó su contenido de tal manera que pudiera ser aplicado en la realización del proyecto final de la asignatura de Programación en la Web, haciendo de este un recurso didáctico y de gran ayuda para la construcción de conocimiento en el estudiante.

En la presentación en clase de la guía básica de la metodología Scrum, tanto el profesor como los estudiantes evaluaron de manera positiva la estructura de la guía y su contenido, asegurando que es un material de apoyo valioso complementario para la asignatura de Programación en la Web.

7. Resultados

En esta sección se presentan los resultados obtenidos para cada una de las tres partes principales del proyecto:

7.1 Aplicativo web clínico

Se cumplió el objetivo de implementar el aplicativo web clínico, basado en su anterior versión, brindando nuevas funcionalidades y actualizando versiones en el stack de sus herramientas. Adicional a esto, se evidenció una mayor complejidad de la esperada al momento de la construcción del aplicativo, dado que su parte frontend fue necesariamente reconstruirla desde cero, haciendo el desarrollo más prolongado que lo pensado inicialmente, a pesar de esto, se logra terminar con una versión funcional de este y acorde a los requerimientos planteados.

7.2 Curso Programación en la Web

Se alcanzó el objetivo de estructurar, organizar y planificar el modelo de curso para la asignatura de programación en la web, cumpliendo con los contenidos requeridos de la asignatura (PostgreSQL, Express, Angular, NodeJS). También se logró con éxito la creación de actividades correspondientes para cada semana del semestre académico, fomentando la creatividad de los estudiantes e impulsando sus habilidades para el desarrollo web.

7.3 Guía básica de la metodología Scrum

Se logró el objetivo de diseñar y crear la guía básica de la metodología Scrum, cumpliendo el requisito de establecer un documento en el que los estudiantes puedan apoyarse para introducirse en el aprendizaje de Scrum y facilitar el desarrollo del proyecto final de la asignatura. También se validó el contenido de la guía al socializarla en la sesión de clase presencial, en donde se explicó a detalle y con éxito ante los estudiantes y el profesor.

8. Conclusiones

El proyecto se culmina de manera exitosa al cumplir con el objetivo general planteado, dado que la nueva versión del modelo de curso para la asignatura de “Programación en la Web” cuenta con contenido actualizado en el stack de herramientas PEAN (PostgreSQL, Express, Angular, NodeJs), también cuenta con nuevas actividades y el reemplazo de algunos temas y actividades anticuadas por tendencias actuales en desarrollo web, todo lo anterior realizado en las plataformas de aprendizaje virtual Moodle y Meiweb.

La nueva versión del aplicativo clínico logra implementar funcionalidades nuevas, haciendo que el uso de este sea más completo, dado que incluye más funcionalidades para los diferentes roles que se tienen dentro de la aplicación, haciendo de esta más robusta y brindando una experiencia de uso más completa y cercana a la realidad.

De manera exitosa se logra la creación de una guía metodológica para que los estudiantes puedan seguir fácilmente el marco de trabajo ágil Scrum. Esta guía les ayudará a saber cómo organizarse y distribuirse diferentes roles dentro de un equipo, fortaleciendo su trabajo en conjunto con otras personas. Con esta guía se mejora el proceso de desarrollo en su proyecto final de asignatura, ya que Scrum lo optimiza y estructura desde el comienzo.

9. Recomendaciones

En cuanto al aplicativo web clínico, como trabajo complementario de los estudiantes se les invita a seguir con la implementación de nuevas funcionalidades que hagan de esta una aplicación más robusta; por ejemplo, se sugiere implementar la funcionalidad para el agendamiento de citas el cual haga más sencillo el proceso para el usuario a la hora de agendar un examen. También se sugiere guardar más información acerca de los usuarios, esto con el fin de brindar una mejor experiencia al momento de clasificarlos, predecir posibles enfermedades, y recomendarles exámenes a realizarse.

Con respecto al modelo de curso, seguir manteniéndolo actualizado, con información de las últimas tendencias en desarrollo web, puesto que es un área que está en constante cambio y crecimiento, lo que hace indispensable mantenerse al día en el uso de sus tecnologías y prácticas. Finalmente explorar la posibilidad de estudiar el contenido de otros stacks de herramientas en desarrollo web para ampliar el conocimiento en los estudiantes.

Referencias Bibliográficas

- Wikipedia contributors. (s/f). *Tim Berners-Lee*. Wikipedia, The Free Encyclopedia.
https://es.wikipedia.org/w/index.php?title=Tim_Berners-Lee&oldid=158808635
- Castelán, J. (2022, julio 26). ¿Cómo ha evolucionado el diseño web en los últimos años? *Talently Blog*. <https://talently.tech/blog/como-ha-evolucionado-el-diseno-web/>
- Qué es MVC*. (2014, enero 2). Desarrolloweb.com. <https://desarrolloweb.com/articulos/que-es-mvc.html>
- Comunicación, D. (2019, julio 26). ¿*Qué es Backend y Frontend?* Descubre Comunicación. <https://descubrecomunicacion.com/que-es-backend-y-frontend/>
- Deyimar, A. (2020, septiembre 9). ¿*Qué es Angular y cuáles son sus ventajas?* Tutoriales Hostinger. <https://www.hostinger.com/co/tutoriales/que-es-angular>
- Gómez, P. (2022, diciembre 20). ¿*Qué es el Backend?* DevCamp. <https://devcamp.es/que-es-backend/>
- Qué es Node.js y por qué debería usarlo*. (2021, mayo 13). Kinsta®; Kinsta. <https://kinsta.com/es/base-de-conocimiento/que-es-node-js/>
- (S/f). Dongee.com. Recuperado el 9 de marzo de 2025, de <https://www.dongee.com/tutoriales/que-es-y-para-que-sirve-express-js/>
- Castellanos, E. (2021, febrero 14). *Git vs GitHub – ¿Qué 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/>

(S/f-b). Amazon.com. Recuperado el 9 de marzo de 2025, de

<https://aws.amazon.com/es/what-is/debugging/#:~:text=LaPrueba%20depuraci%C3%B3n%20y%20las%20pruebas%20sobre%20procesos%20complementarios%20que%20garantizan,para%20identificar%20fallos%20y%20errores>

¿Qué es la Inteligencia Artificial (IA)? (2024, junio 14). *Ibm.com*. <https://www.ibm.com/mx-es/topics/artificial-intelligence>

¿Qué es una base de datos? (2020, noviembre 24). Oracle.com; Oracle.

<https://www.oracle.com/co/database/what-is-database/>

Herrera, A. (2021, julio 23). *Qué es el eLearning*. Innovación y Cualificación.

<https://www.innovacionycualificacion.com/plataforma-elearning/que-es-elearning/>

¿Qué es End-to-End Testing? (2018, octubre 31). <https://platzi.com/clases/1411-testing-cypress/14608-que-es-end-to-end-testing/>

Cypress testing solutions. (s/f). Cypress.io. Recuperado el 9 de marzo de 2025, de

<https://docs.cypress.io/app/get-started/why-cypress>

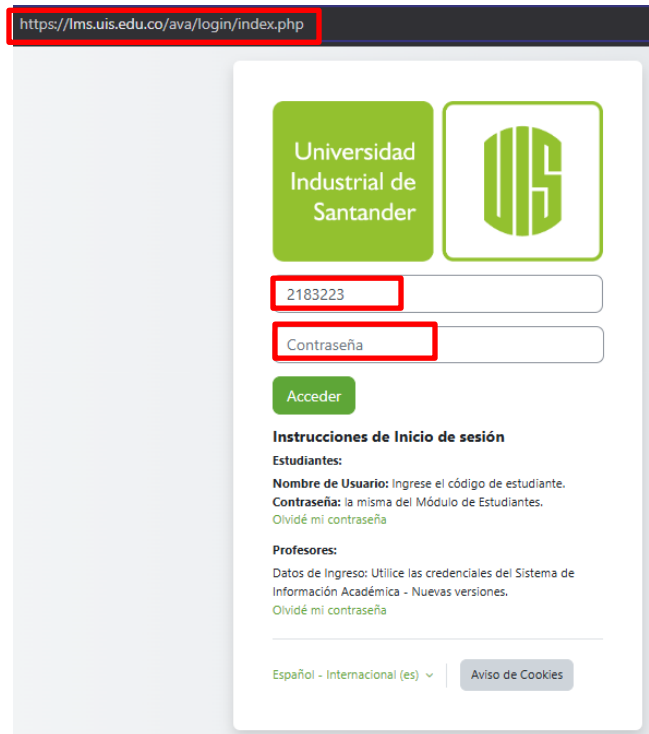
(S/f). Amazon.com. Recuperado el 9 de marzo de 2025, de <https://aws.amazon.com/es/what-is/unit-testing/>

Apéndices

Apéndice A. Manual de usuario para el uso del curso virtual de Programación en la Web en la plataforma Moodle.

Para acceder al curso virtual primero hay que iniciar sesión en la plataforma de la siguiente manera:

Vista inicio de sesión en Moodle



https://lms.uis.edu.co/ava/login/index.php

Universidad Industrial de Santander

2183223

Contraseña

Acceder

Instrucciones de Inicio de sesión

Estudiantes:
Nombre de Usuario: Ingrese el código de estudiante.
Contraseña: la misma del Módulo de Estudiantes.
[Olvidé mi contraseña](#)

Profesores:
Datos de Ingreso: Utilice las credenciales del Sistema de Información Académica - Nuevas versiones.
[Olvidé mi contraseña](#)

Español - Internacional (es) | [Aviso de Cookies](#)

El usuario debe ingresar a la anterior dirección https señalada en la imagen. Posteriormente debe ingresar las credenciales de usuario y contraseña. El usuario corresponde al código de estudiante y la contraseña será la misma con la que el estudiante ingresa al sistema de estudiantes UIS.

De ingresar correctamente, al usuario se le mostrará la siguiente pantalla, en donde podrá encontrar los cursos en los que se encuentra actualmente matriculado y entre ellos claro, estará el curso de Programación en la Web.

Vista home en Moodle



Al dar clic en el curso de Programación en la Web, al usuario se le mostrará la siguiente pantalla de bienvenida en la que encontrará los aspectos principales del curso.

Vista curso programación en la web en Moodle



Entre las secciones principales del curso se encuentran:

Introducción. En donde el usuario podrá encontrar información en general acerca del curso, temas que se tratarán, etc...

*Vista introducción en curso programación en la web en Moodle***Introducción****PROGRAMACIÓN EN LA WEB**

Código: 22967

Centic:

profesor: Manuel Guillermo Flórez Becerra
 oficina: 221 LP (Edificio Laboratorios Pesados)
 Grupo Investigación y Desarrollo: GD-CONUSIS

Horario Consulta: Por definir

Programación en la web usando (Postgresql Express Angular Nodejs)

UNIVERSIDAD INDUSTRIAL DE SANTANDER FACULTAD DE INGENIERÍAS FÍSICOMECÁNICAS ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA Programa Ingeniería de Sistemas	
PROGRAMACIÓN EN LA WEB	
CÓDIGO: 22967	NÚMERO DE CRÉDITOS: 4
REQUISITOS:	
INTENSIDAD HORARIA SEMANAL: 12 HORAS	TAD: 4 HORAS TI: 8 HORAS
TALLERES: 4	LABORATORIO: 4 TEÓRICA: 4
JUSTIFICACIÓN	

Objetivos y políticas. En donde el usuario podrá encontrar información relacionada como las políticas del curso, estrategias de evaluación o competencias a desarrollar.

*Vista objetivos y políticas en curso programación en la web en Moodle***Objetivos y Políticas****PROGRAMACIÓN EN LA WEB USANDO PEAN STACK código: 22967**

Horas semanales 2T y 2P
 Prerrequisitos
 Área Profesional
 Dependencia Escuela de Informática y Sistemas

JUSTIFICACIÓN

En la actualidad, las redes de computadoras desempeñan un papel fundamental en la forma en que vivimos y trabajamos. Este es un momento crucial en la evolución de la tecnología, especialmente en lo que representa a la transmisión de información a través de estas redes. Además, se observa un patrón de comportamiento en el que la demanda de servicios informáticos se centra en la creación de sitios web cada vez más avanzados utilizando frameworks y bibliotecas sólidas, así como en la programación avanzada en línea.

Para preparar a los futuros profesionales de Ingeniería de Sistemas y hacerlos altamente competitivos en el mercado laboral, es esencial que adquieran una base teórica sólida. Esta base teórica les servirá como una tarjeta de presentación en sus entornos laborales, donde necesitarán aplicar conceptos avanzados en el análisis, diseño y desarrollo de sistemas de software.

PROPÓSITOS

- o Formar al estudiante con capacidades y habilidades para comprender el entorno de desarrollo Web usando tecnologías como Nodejs-Express-Angular-Postgresql.
- o Ofrecer una respuesta para mejorar la creación en entornos en línea y respaldar esta propuesta con argumentos sólidos.
- o Conocer las herramientas para el diseño y programación de aplicaciones WEB.

COMPETENCIAS

El estudiante:

- o Aborda temas relacionados con sistemas de información diseñados para entornos web.
- o Formula propuestas destinadas a mejorar el proceso de desarrollo en el contexto de internet.
- o Justifica y respalda las soluciones propuesta en el ámbito del desarrollo web con argumentos sólidos.
- o Evalúa críticamente las soluciones propuesta, considerando posibles desafíos y riesgos asociados.

Cronograma de evaluaciones. En donde el usuario podrá encontrar la planeación de evaluaciones y actividades a realizar durante el desarrollo del curso.

Vista cronograma de evaluaciones en curso programación en la web Moodle

Cronograma de Evaluaciones

PROGRAMACIÓN EN LA WEB I-2025		
ACTIVIDADES	%	FECHAS
Tareas, quizzes, Talleres en clase, ensayos 1. Participación y Preguntas en clase 2. Quiz (Generalidades- HTML- CSS) 3. Trabajo grupal (diseño e implementación una plantilla web: 6 semanas despues de inicio).	20%	Inicio Clases: Febrero 3 Marzo domingo 23 Registro primera nota en el sistema (Aprox 7 semanas despues de inicio)
Plan proyecto semestre fecha limite: 4 semanas despues de inicio de clases	30% Plan: 15%	Marzo Lunes 3 El formato del plan lo encuentra en: me/web0>Contenidos>plan_proyecto_semestre
FASE I proyecto (lado del cliente) (Aprox 5 semanas despues del plan) Esquema del proyecto e implementación de la interfaz gráfica aplicando HTML5, CSS, JavaScript, JQuery, bootstrap, React) 80%	Fase I: 85%	Abril Jueves 24 fecha limite para subir el informe el .zip o todo en github La socialización se realizará durante las clases de la semana correspondiente
FASE II proyecto (lado del servidor + lado del cliente) (última semana de clases) lado del servidor: Implementación en nodeJS, Express; los scripts de la lógica del negocio y conexiones y manejo de la base de datos Mysql) o MongoDB según requisitos que especifique el profesor (100%)	30%	Mayo Lunes 26 Fecha limite para subir el .zip Subir documentación (Todo en un .zip) o todo en github Sustentación en el Aula Cada grupo dispone de 1 hora para realizar exposición usando la plataforma zoom.

El planificador del curso. En donde el usuario podrá encontrar la planificación por semanas para los temas a ver durante el curso, las actividades, y las competencias a desarrollar semanalmente.

Vista planificador en curso programación en la web en Moodle

Introducción	Objetivos y Políticas	Cronograma de Evaluaciones	Planificador	Materiales y recursos	Extras					
Planificador	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7	Semana 8	Semana 9	Semana 10
Semana 11	Semana 12	Semana 13	Semana 14	Avances del proyecto final	Presentación y calificación del proyecto final					

Semana 1

Competencias teóricas

- o Presentación general del curso.
- o Fundamentos de la programación en la web
- o Fundamentos de Internet y el modelo cliente servidor
- o Sistema de control de versiones Git y sus comandos principales.
- o Marco de trabajo ágil Scrum

Competencias prácticas

- o Conocer el contenido de la asignatura su forma de evaluación y las políticas del curso.
- o Comprender los conceptos básicos de la web.
- o Realizar una Introducción a la Web, Conocer terminología Web
- o Descarga e instalación del editor de texto Visual Studio code
- o Inicializar repositorios y realizar seguimiento de cambios
- o Asumir roles de Scrum (Product Owner, Scrum Master, Equipo de Desarrollo) y colaborar de manera efectiva en un entorno autoorganizado.
- o Crear y gestionar un Product Backlog priorizando elementos según objetivos del proyecto.
- o Participar en la planificación, ejecución y seguimiento de tareas durante un sprint (Sprint Planning, Daily Scrum).
- o Evaluar entregables en una Sprint Review y proponer mejoras en una Sprint Retrospective.

Actividades

- Primera clase:

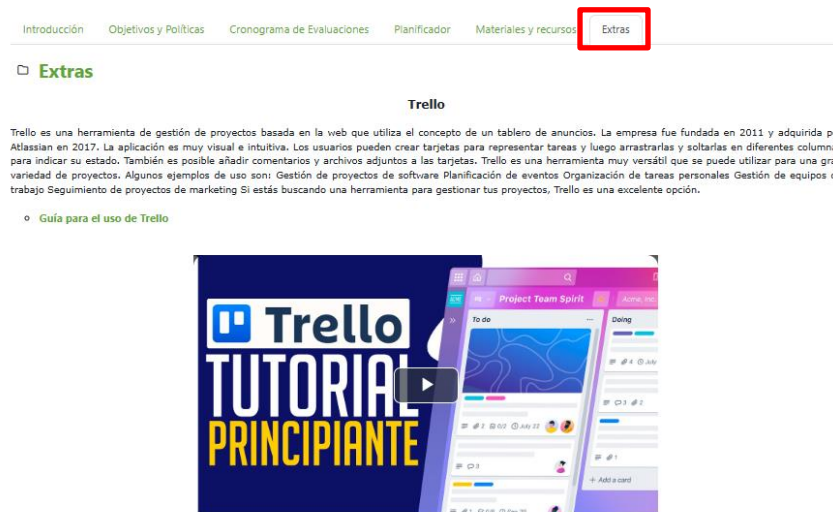
Los materiales y recursos. En esta sección, el usuario encontrará el material de las actividades a realizar y los recursos en distintas presentaciones como vídeos, enlaces y prácticas que serán apoyo durante el desarrollo del curso. Esta sección está dividida por semanas para facilitar la organización del contenido.

Vista contenido en curso programación en la web en Moodle



Extras. Para finalizar, está la sección de extras, en la que el usuario podrá encontrar enlaces, vídeos o actividades de apoyo que, a pesar de no ser indispensables, son beneficiosas para el curso, complementando su ruta de aprendizaje y enriqueciendo el conocimiento del usuario.

Vista extras en curso programación en la web en Moodle



Apéndice B. Manual de usuario para el uso del curso virtual de Programación en la Web en la plataforma Meiweb.

Para acceder al curso, hay que iniciar sesión en la plataforma de la siguiente manera:

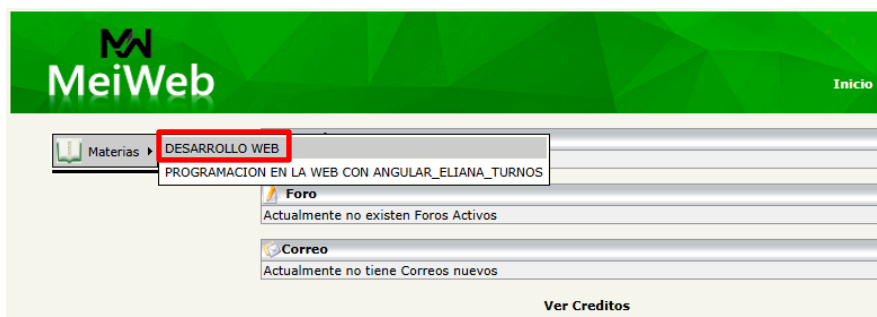
Vista inicio de sesión en Meiweb



El usuario debe ingresar a la anterior dirección https señalada en la imagen. Posteriormente debe ingresar las credenciales de usuario y contraseña que el usuario ha creado anteriormente al registrarse en la plataforma Meiweb.

De ingresar, al usuario se le mostrará la siguiente pantalla, y al dar clic en Materias, estará el curso de Programación en la Web.

Vista home en Meiweb



Al dar clic en el curso de Programación en la Web, al usuario se le mostrará la siguiente pantalla de bienvenida en la que encontrará los aspectos principales del curso:

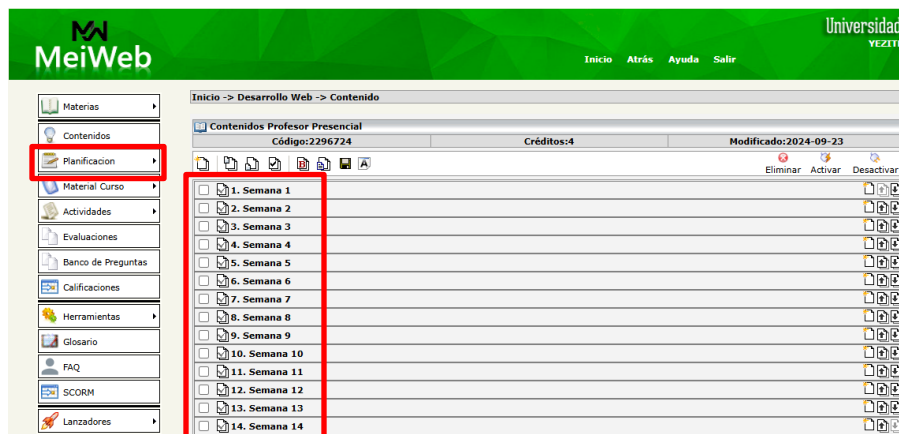
Vista home en curso programación en la web en Meiweb

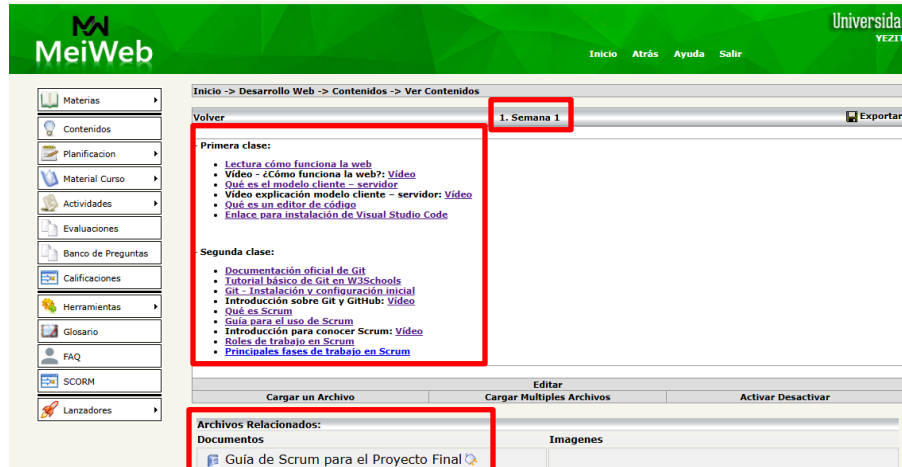


Entre las secciones principales del curso se encuentran:

Contenidos. En esta sección el usuario podrá encontrar el material correspondiente al contenido a ver durante todo el desarrollo del curso. Estos contenidos están divididos por semanas y en cada una de estas el usuario podrá encontrar material como vídeos, enlaces y documentos con actividades a realizar.

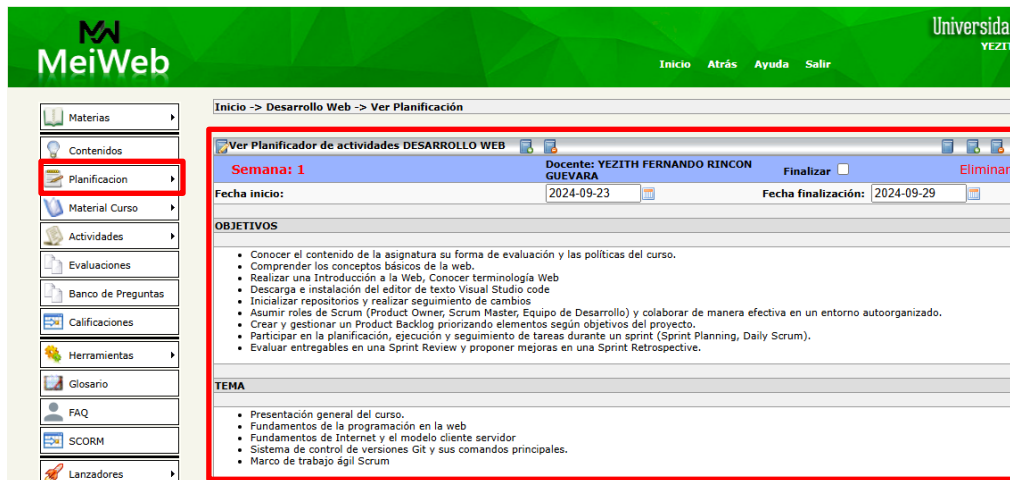
Vistas de la sección contenidos en curso programación en la web en Meiweb





Planificador. En esta sección el usuario podrá encontrar la planificación por semanas del contenido, temas y actividades a realizar durante el desarrollo del curso, esto con el fin de facilitar la organización del curso y orientar al usuario semana a semana.

Vista planificador en curso programación en la web en Meiweb



Material Curso. En esta sección el usuario podrá encontrar el material y contenido del curso organizado por semanas. Esta es una sección para poner a disposición del usuario otra manera de organizar y poder visualizar el material de todo el curso que puede consistir en ejercicios, guías de actividades y demás.

Vista material curso en curso programación en la web en Meiweb

The screenshot shows the MeiWeb interface for the course 'Desarrollo Web'. The top navigation bar includes 'Inicio', 'Atrás', 'Ayuda', and 'Salir'. The left sidebar contains a menu with items like 'Materias', 'Contenidos', 'Planificación', 'Material Curso' (highlighted with a red box), 'Actividades', 'Evaluaciones', 'Banco de Preguntas', 'Calificaciones', 'Herramientas', 'Glosario', 'FAQ', 'SCORM', and 'Lanzadores'. The main content area is titled 'Inicio : Desarrollo Web : Biblioteca' and shows a list of course materials for 'DESARROLLO WEB' sorted by topic. The materials are organized by week (Semana 1 to Semana 13) and include files such as 'Guia...', 'postg...', 'auth...', 'node_...', 'expre...', 'guard...', 'routi...', 'react...', 'reque...', 'input...', and 'hooks...'.

Inicio : Desarrollo Web : Biblioteca	
Los archivos estan ordenados por: Tema	
DESARROLLO WEB	
Semana 1	Guia...
Semana 13	postg... auth...
Semana 12	node_... expre...
Semana 11	guard... routi...
Semana 10	react... reque...
Semana 9	input... hooks...

Apéndice C. Manual de usuario para el uso del aplicativo clínico.

A continuación, se mostrará un instructivo para usuarios con rol de Paciente. Se espera que, con este, el usuario entienda cuáles son las acciones que puede realizar y cómo realizarlas.

Al iniciar, el aplicativo mostrará la siguiente pantalla inicial, con información del laboratorio.

Vista home lab clínico



Para hacer uso del aplicativo, lo primero es crear una cuenta en caso de no tenerla, el usuario debe llenar el registro con sus datos personales como documento de identidad, nombre, apellidos, fecha de nacimiento, correo electrónico y debe crear una contraseña que cumpla con los parámetros establecidos que son: tener mínimo una minúscula y mayúscula, tener más de 8 caracteres, tener algún carácter especial como (*,.,+) y tener un número.

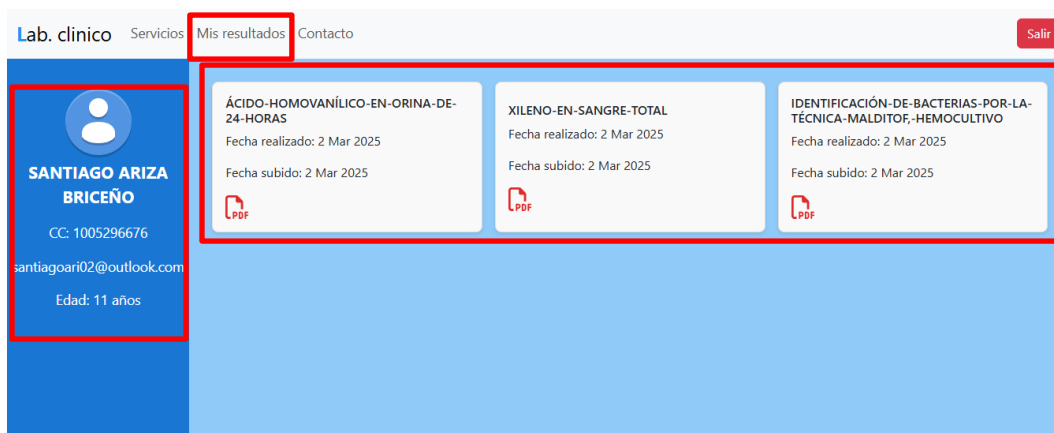
Vista registro en lab clínico

Luego de crear la cuenta, el usuario podrá iniciar sesión, luego de iniciar sesión correctamente, como pantalla inicial, el usuario visualizará un panel izquierdo que contiene información personal, y a la derecha verá la sección Mis resultados, en donde estarán los resultados disponibles para descargar.

Vista inicio de sesión en lab clínico

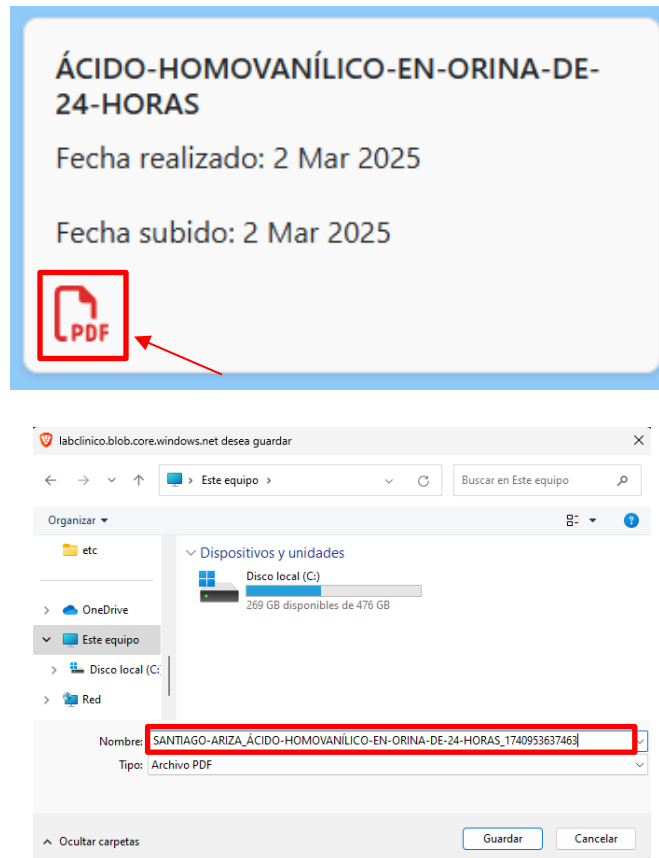


Vista mis resultados en lab clínico



Al dar clic en el ícono de PDF del resultado que elija, el usuario podrá descargar el resultado y ver el contenido del mismo.

Vistas botón para descargar resultado en lab clínico



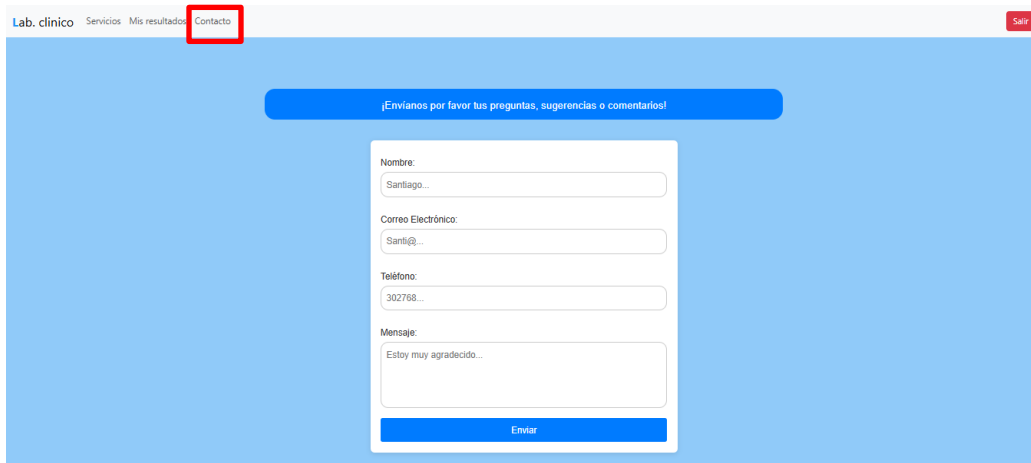
En la sección de Servicios, el usuario podrá encontrar la información relacionada con cada uno de los exámenes que ofrece el laboratorio, además podrá leer una descripción breve acerca de cada uno de estos.

Vista servicios en lab clínico



En la sección de Contacto, el usuario podrá realizar sus PQR'S o comentarios orientados a la mejora del laboratorio.

Vista contacto en lab clínico



Si el usuario ingresa por error a una ruta no existente, se le mostrará la siguiente pantalla 404, notificándole que no se pudo encontrar la ruta. En esta pantalla, el usuario también podrá encontrar un botón para volver al inicio.

Vista página de error o 404 en lab clínico

