



Universidad Tecnológica ECOTEC

Facultad de Ingenierías

Título del trabajo:

Desarrollo de un sistema informático para la gestión y monitoreo de lectura, mediante un sistema web y aplicación móvil, para la Fundación Leonidas Ortega.

Línea de Investigación:

Tecnologías de la Información y la Comunicación

Modalidad de titulación:

Propuesta Tecnológica

Carrera:

Ingeniería en sistemas énfasis en sistema

Título a obtener:

Ingeniero en Sistemas

Autor (a):

Cesar Josue Vargas Sosa

Tutor (a):

Ing. Luis Espinoza

Samborondon – Ecuador

2022

Dedicatoria

Este logro está dedicado a Dios principalmente, por guiarme en esta trayectoria para alcanzar mi objetivo y a mis padres que han sido mi apoyo incondicional en este trayecto y que sin ellos no pudiera haber logrado culminar este objetivo propuesto y les prometo ser un excelente profesional actuando siempre de la mano de Dios.

Le dedico también este logro al Dr. Carlos Ortega Maldonado por haberme dado la oportunidad de poder tomar esta increíble carrera en la prestigiosa Universidad Ecotec.

Agradecimiento

Primero agradezco a Dios por haberme dado salud y la familia que me ha puesto. Agradezco a mis padres por siempre estar para mí en este largo trayecto para poder alcanzar mi objetivo de ser un profesional, gracias por todos los consejos, conocimiento, paciencia, horas de sueño, ideas y muchas cosas más por las que puedo agradecerles, mis padres son mi más grande inspiración que puedo tener, por todo eso y más les agradezco y les expreso que los amo mucho.

Agradezco al Dr. Carlos Ortega Maldonado por creer en mí y ayudarme con una beca y poder cumplir mi sueño de tener la oportunidad de graduarme como ingeniero en la Universidad Ecotec.

Y por último agradecer a la Universidad Ecotec por los excelentes maestros que tienen y que compartieron su conocimiento conmigo para formarme como un excelente profesional.

ANEXO 14

CERTIFICADO DE APROBACION DEL TUTOR PARA LA PRESENTACION A REVISION DEL TRABAJO DE TITULACION

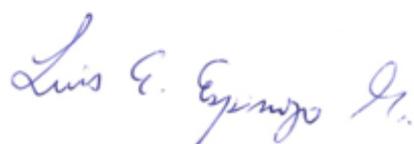
Samborondón 26 de Octubre de 2022

Magíster
Ing. Erika Ascencio Jordán, Mgs.
Decano(a) de la Facultad
Ingeniería
Universidad Tecnológica ECOTEC

De mis consideraciones:

Por medio de la presente comunico a usted que el trabajo de titulación TITULADO: DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN Y MONITOREO DE LECTURA, MEDIANTE UN SISTEMA WEB Y APLICACIÓN MÓVIL, PARA LA FUNDACIÓN LEONIDAS ORTEGA. Según su modalidad PROPUESTA fue revisado, siendo su contenido original en su totalidad, así como el cumplimiento de los requerimientos establecidos en la guía para la elaboración del trabajo de titulación, Por lo que se autoriza a: **Vargas Sosa Cesar Josue** para que proceda a su presentación para la revisión de los miembros del tribunal de sustentación.

ATENTAMENTE,



Luis Enrique Espinoza Mendoza

Tutor(a)

ANEXO 15

Habiendo sido nombrado **Luis Enrique Espinoza Mendoza** tutor del trabajo de titulación "DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN Y MONITOREO DE LECTURA, MEDIANTE UN SISTEMA WEB Y APLICACIÓN MÓVIL, PARA LA FUNDACIÓN LEONIDAS ORTEGA" elaborado por Cesar Josue Vargas Sosa, con mi respectiva supervisión como requerimiento parcial para la obtención del título de Ingeniero en sistemas.

Se informa que el mismo ha resultado tener un porcentaje de coincidencias 9 (%) mismo que se puede verificar en el siguiente link:

<https://secure.orkund.com/old/view/142013995-108379-527671#Dcc5DslwFEDBu7h+QvbfnsqChQBckGalli7w3Tzae+zbVfpyEAMcaTQig5U0ED/ndjABDPMscQK7/jABVfccMcDT3wSnRBCCSOSKKGKSqxRSSOdDDJvtHO9jvVc+/3YH23rl16zas4oK3Md4t8f>.

Adicional se adjunta print de pantalla de dicho resultado.

The screenshot displays the URKUND interface. On the left, a sidebar shows document details: 'Documento: tesis_josueFinal.docx (D148994917)', 'Presentado: 2022-11-07 17:29 (-05:00)', 'Presentado por: Lespinoza (lespinoza@ecotec.edu.ec)', 'Recibido: lespinoza.ecotec@analysis.orkund.com', and 'Mensaje: Mostrar el mensaje completo'. The main content area shows a table of sources with columns 'Lista de fuentes' and 'Bloques'. The table lists 'UNIVERSIDAD TECNOLÓGICA ECOTEC / tesis_j...' and 'UNIVERSIDAD TECNOLÓGICA ECOTEC / Xavier...'. A progress bar indicates '9%' of the document is text present in the sources. Below the table, a list of document sections is visible, including 'procesos actuales y del sistema 40 4.4', 'Descripción del proyecto 44 4.5.2', 'Requerimientos 45 4.6', 'Arquitectura basada en componentes', and 'Arquitectura cliente servidor 60 4.8.5'.

FIRMA DEL TUTOR:

Luis E. Espinoza M.

ANEXO 16

CERTIFICADO DE APROBACIÓN DEL TUTOR PARA LA PRESENTACIÓN DEL TRABAJO DE TITULACIÓN CON INCORPORACIÓN DE LAS OBSERVACIONES DE LOS MIEMBROS DEL TRIBUNAL

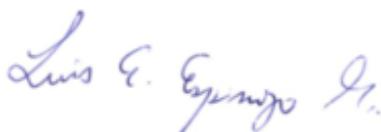
Samborondón 07 de Noviembre de 2022

Magíster
Ing. Erika Ascencio Jordán, Mgs.
Decano(a) de la Facultad
Ingeniería
Universidad Tecnológica ECOTEC

De mis consideraciones:

Por medio de la presente comunico a usted que el trabajo de titulación TITULADO: DESARROLLO DE UN SISTEMA INFORMÁTICO PARA LA GESTIÓN Y MONITOREO DE LECTURA, MEDIANTE UN SISTEMA WEB Y APLICACIÓN MÓVIL, PARA LA FUNDACIÓN LEONIDAS ORTEGA según su modalidad PROPUESTA TECNOLÓGICA; fue revisado y se deja constancia que el estudiante acogió e incorporó todas las observaciones realizadas por los miembros del tribunal de sustentación por lo que se autoriza a: **Vargas Sosa Cesar Josue**, para que proceda a la presentación del trabajo de titulación para la revisión de los miembros del tribunal de sustentación y posterior sustentación.

ATENTAMENTE,



Ing. Luis Enrique Espinoza Mendoza

Tutor(a)

Tabla de contenido

Tabla de contenido	6
Resumen	11
Abstract	12
Introducción	13
Contexto histórico	14
Antecedentes.....	15
Planteamiento del problema científico	16
Pregunta problemática.....	17
Límite temporal de investigación.....	17
Límites espaciales de la investigación	17
Objetivo general	17
Objetivos específicos.....	17
Justificación	18
Alcance de la Investigación	18
Novedad o aspecto innovador	19
1 Capítulo I: Marco Teórico	21
1.1 Metodología ágil de desarrollo SCRUM	21
1.2 Metodología en cascada	22
1.3 Metodología evolutiva	23
1.4 Modelo Espiral	23
1.5 Modelo por prototipo	24
1.6 Programación orientada a objetos	25
1.7 Patrón de diseño.....	25

1.8	Arquitectura cliente servidor.....	26
1.9	Tecnologías de desarrollo de software.....	27
1.10	Bases de datos relacionales	27
1.11	Autenticación de usuarios	28
1.12	Selección de metodologías	29
2	Capítulo II: Marco Metodológico	31
2.1	Enfoque de investigación	31
2.2	Identificación y definición de variables	31
2.2.1	Identificación de variables.....	31
2.2.2	Definición de variables.....	32
2.2.3	Operacionalización de las variables.....	32
2.3	Universo y muestra	33
2.3.1	Universo.....	33
2.3.2	Muestra	33
2.4	Métodos empíricos y estadísticos	33
3	Capítulo III: Análisis e Interpretación De Los Resultados	36
4	Capítulo IV: Propuesta	40
4.1	Ingeniería de software	40
4.2	Escenario actual	40
4.3	Diagrama lineal de procesos actuales y del sistema	42
4.4	Aplicación de metodología	46
4.5	Comunicación	46
4.5.1	Descripción del proyecto.....	46
4.5.2	Alcance del proyecto.....	46
4.5.3	Requerimientos	47
4.6	Planeación	51
4.6.1	Estimación de tiempos.....	51
4.6.2	Análisis de riesgo.....	51
4.6.3	Matriz de riesgos	52

4.7	Modelado.....	53
4.7.1	Análisis.....	53
4.8	Diseño	56
4.8.1	Prototipos.....	56
4.8.2	Diagrama de casos de uso.....	60
4.8.3	Arquitectura basada en componentes y Widgets.....	61
4.8.4	Arquitectura cliente servidor	62
4.8.5	Diccionario de datos	62
4.8.6	Modelo entidad relación	65
4.9	Construcción.....	65
4.9.1	Código.....	65
4.9.2	Prueba	67
4.10	Despliegue.....	73
4.10.1	Entrega.....	73
	Conclusión.....	75
	Recomendaciones	77
	Bibliografía.....	79
	Anexos.....	82

Índice De Tablas

Tabla 1:	Funcionalidad de las variables	32
Tabla 2:	Situación actual de listado de lector. Fuente: elaboración propia (2022)	41
Tabla 3:	Situación actual de listado de libros. Fuente: elaboración propia (2022)	41
Tabla 4:	Autenticación de acceso de administrador.	47
Tabla 5:	Autenticación de acceso de lector.	47
Tabla 6:	Mantenimiento de listado de lectores.	48
Tabla 7:	Mantenimiento de listado de libros.	48

Tabla 8: Módulo de préstamos.....	48
Tabla 9: Módulo de comentarios.....	48
Tabla 10: Visualización de estadísticas.....	49
Tabla 11: Listado de préstamos solicitados	49
Tabla 12: Requerimientos no funcionales	50
Tabla 13: Matriz de riesgos. Fuente: propia (2022)	52
Tabla 14: Descripción del producto.....	53
Tabla 15: Viabilidad técnica	55

Índices De Figuras

Ilustración 1: Modelo Scrum.....	22
Ilustración 2: Modelo en Cascada	22
Ilustración 3: Ciclo de vida del modelo espiral. Fuente: (Pressman, 2010).....	24
Ilustración 4: Privilegios de una tabla. (Carvajal, Manuel, León, & Alina, 2011).....	28
Ilustración 5: Diagrama lineal de préstamo de libros. Fuente: Elaboración propia (2022).	43
Ilustración 6: Diagrama lineal de Monitoreo de lectura. Fuente: Elaboración propia (2022).	44
Ilustración 7: Diagrama lineal de Análisis de datos. Fuente: Elaboración propia (2022).	45
Ilustración 8: Administrador web. Fuente: propia (2022).	57
Ilustración 9 Prototipo de Login. Fuente: Propia (2022)	58
Ilustración 10: Perfil Móvil Fuente: Propia (2022)	59
Ilustración 11: Diagrama de casos de uso	60
Ilustración 12: Modelo de base de datos.....	65
Ilustración 13: Controlador de lector. Fuente: propia (2022).....	66
Ilustración 14: Estructura de carpeta de Nest js y Angular	67
Ilustración 15 Archivo spec de Lector service. Fuente: Propia (2022)	72
Ilustración 16: Karma v3.1.6 Fuente: Propia (2022)	73
Ilustración 17: Reunión sobre prototipos	83
Ilustración 18: Reunión sobre requerimientos	84
Ilustración 19: Reunión sobre validaciones de la app.....	85

Ilustración 20: Reunión sobre funcionalidad de app	86
Ilustración 21 Login del administrador. Fuente: Propia (2022)	87
Ilustración 22 Modulo de Estadísticas Generales y específicas. Fuente: Propia (2022)	87
Ilustración 23 Modulo Estadísticas tabla de monitoreo. Fuente: Propia (2022)	88
Ilustración 24 Funcionalidad préstamo libro. Fuente: Propia (2022)	88
Ilustración 25 Modulo Libro. Fuente: Propia (2022)	89
Ilustración 26 Funcionalidad crear libro. Fuente: Propia (2022)	89
Ilustración 27 Modulo Lector. Fuente: Propia (2022)	90
Ilustración 28 Funcionalidad agregar lector. Fuente: Propia (2022)	90
Ilustración 29 Modulo comentarios Fuente: Propia (2022)	91
Ilustración 30 Login App Móvil Fuente: Propia (2022)	92
Ilustración 31 Perfil Lector. Fuente: Propia (2022)	93
Ilustración 32 Funcionalidad comentar libro Fuente: Propia (2022)	94

Resumen

El presente proyecto se realizó con la finalidad de monitorear y controlar el proceso de lectura mediante una aplicación web y móvil, las cuales funcionan como herramientas para gestionar los préstamos que hace la Fundación Leonidas Ortega hacia el lector. Cabe destacar, que el desarrollo de este proyecto se usó metodologías y arquitecturas que facilitan el desarrollo ágil. El denominado arquitectura cliente servidor, modelos basados en componentes, módulos y metodología Espiral combinada con prototipo fue de gran ayuda para que el desarrollo del software sea eficiente. No obstante, se realizó una exhaustiva investigación sobre los procesos que maneja la Fundación Leonidas Ortega en el sector bibliotecario y su control actual. Según lo mencionado, se realizó un análisis detallado y definir prototipos y alcance del proyecto que se adecue de la mejor forma a la Fundación. Cabe mencionar que la tecnología evoluciona y es importante que las organizaciones se adapten a los cambios tecnológicos, optar por un sistema es una opción favorable para la Fundación debido a la optimización de procesos que este brinda.

Abstract

This project was carried out with the purpose of monitoring and controlling the reading process through a web and mobile application, which will function as tools to manage the loans that “Fundación Leonidas Ortega” makes to the reader. It should be noted that the development of this project used methodologies and architectures that facilitated agile development. The so-called client architecture server, component-based models, modules and Spiral methodology combined with prototyping was of great help in making the software development efficient. However, an exhaustive investigation was carried out on the processes handled by “Fundación Leonidas Ortega” in the library sector and its current control. As mentioned, a detailed analysis was carried out and prototypes and scope of the project that best suits the Foundation were defined. It is worth mentioning that technology evolves and it is important for organizations to adapt to technological changes. Opting for a system is a favorable option for the Foundation due to the optimization of processes that it provides.

Introducción

El desarrollo de las tecnologías aporta al campo de la educación extraordinarias posibilidades para potencializar y amplificar su incidencia en los procesos de enseñanza aprendizaje, procesos que están en permanente evolución, de esa forma la Fundación Leonidas Ortega Moreira, institución social que desde 1988 trabaja en el campo de la Educación, brindando través de becas el acceso a la educación de calidad de estudiantes talentos con vocación de servicio, está impulsando dentro del Programa Liderazgo, un proyecto de promoción de la lectura para sus becarios. Esta iniciativa que busca motivar hábitos lectores en sus estudiantes que no cuenta al momento con soluciones tecnológicas que le permita a la Fundación implementar acciones de registro, seguimiento, monitoreo y análisis de los resultados que busca alcanzar en su ejecución y desarrollo, más allá de los controles manuales que tradicionalmente se aplican.

De acuerdo con lo expresado por la Coordinadora del Programa Liderazgo, para el proyecto de promoción a la lectura, contar con herramientas tecnológicas para el inventario y monitoreo de lectura, que permita llevar un control eficiente de las acciones que se implementan, así como monitorear los indicadores de progreso en los hábitos lectores de nuestros beneficiarios, aportando en el fortalecimiento la lectura como un hábito de aprendizaje para toda la vida, considerando a la lectura como nos dice Sergio Haro (Alcaraz, 2014) como un proceso cognoscitivo que nos permite interpretar los signos gráficos por medio de recreaciones mentales que habilita ver lo que no está presente, se puede concluir en imaginar una realidad. En este contexto la Fundación Leonidas Ortega Moreira a través de estas herramientas llevará el registro, monitoreo y control de los libros que cada becario esté motivado a leer, como resultado de las estrategias implementadas en el desarrollo del proyecto y como un acto consciente y voluntario que contribuirá al desarrollo de sus habilidades personales y profesionales.

Contexto histórico

Con el pasar de los años, los sistemas de información han evolucionado y tomando fuerza en las organizaciones, dejando a un lado los métodos tradicionales donde los registros se llevaban en documentos físicos. Cabe destacar que las empresas optan por las tecnologías de información con motivo de optimizar procesos que se realizan durante el día a día. En el mercado tecnológico ha tomado lugar sistemas informáticos contables, gestión para toma de decisiones, análisis de datos, comercio electrónico, entre otros. No cabe duda que existen infinidad de sistemas informáticos, sin embargo, estos cumplen funcionalidades específicas para una o varias áreas. Cabe destacar que las empresas optan por un software de acuerdo a sus necesidades. En las fundaciones, se puede diferir que lo que necesita es un sistema de gestión para el control de usuarios, la cual suena óptimo. Sin embargo, los sistemas informáticos para el control de lectura no son típicos en el mercado de TI ni en una Fundación, según el contexto histórico, los sistemas informáticos tienden a ocupar otras áreas con diferentes funcionalidades, pero suplir un área de control de lectura resuelve muchos problemas en una Fundación. Es importante destacar que los sistemas de información suelen resolver problemas o necesidades. En las operaciones diarias de una organización suelen ocurrir muchos procesos que con el tiempo son solventados gracias a la tecnología, esto quiere decir que no es descabellado optar por un sistema informático para el control y monitoreo de lectura de una Fundación.

Antecedentes

Según lo dicho por (Viveros, n.d.), su historia empieza en el año 1989 como parte del programa de responsabilidad social, del entonces Banco Continental, con la misión de implementar Proyectos Sociales para beneficiar a los colaboradores de la institución y su familia. Esta institución tiene como base dos objetivos fundamentales las cuales son: Proyectos internos. - Consiste en la creación de espacios culturales para los hijos de sus trabajadores. Proyectos externos. - Consiste en la implementación de 35 bibliotecas en el país, la cual fue una obra realizada con éxito.

La Fundación Leonidas Ortega durante años se destaca por los convenios con instituciones educativas, las cuales son pilares para planes de becas educativas que ayudan a sus estudiantes a decidir por optar por la Fundación como una entidad de crecimiento educativo y profesional. La tecnología no es alejada de la institución debido a que este tiene un sistema que ayuda a los postulantes pasar por un proceso de aprobación y una vez aprobado, los estudiantes tienen acceso a una cuenta con funcionalidades que brinda la plataforma con respecto a notas, datos personales y documentos.

Cabe destacar, que la Fundación Leonidas Ortega se caracteriza en la enseñanza como la verdadera transformación de vidas, la cual ayuda a sus estudiantes a tener un mejor futuro fundamentado en valores y educación. No obstante, la Fundación, destaca por las continuas capacitaciones y talleres brindadas por maestros y profesionales que otorgan un conocimiento extra a los estudiantes. La institución es reconocida como pilar en el reconocimiento de becas a sus estudiantes, otorgando garantías donde ellos puedan destacarse en el grado seleccionado.

Planteamiento del problema científico

La Fundación Leonidas Ortega Moreira para el desarrollo de la Educación y la Cultura, es una organización social sin fines de lucro que desde 1988 viene trabajando en programas sociales que han contribuido a mejorar la educación y la cultura en el país. En este contexto y desde su Programa Liderazgo (Fundación Leonidas Ortega, 2022), la Fundación está implementando el proyecto de animación a la lectura para los beneficiarios de la institución, niños y jóvenes talentosos con limitados recursos que se destacan por sus méritos académicos y culturales. Sin embargo, este proyecto que busca promocionar la lectura, requiere un control del inventario de los libros de lectura que han sido donados a la Fundación, así como el registro de los usuarios que motivados por las estrategias que se implemente en el desarrollo del proyecto, seleccionen los libros para llevarlos a casa y proceder a su lectura, una herramienta para el usuario que pueda registrar en su perfil los libros que está leyendo y quisiera leer, así como compartir su comentario del mensaje que le ha significado la lectura del libro escogido.

Para la Fundación es de vital importancia lograr las condiciones óptimas para la ejecución del proyecto de animación a la lectura, porque la lectura trae consigo algunos beneficios, "que desarrolla en el ser humano, múltiples capacidades de comprensión, interpretación y análisis". La institución necesita monitorear los indicadores de hábitos lectores de los estudiantes becarios, grupo beneficiario del proyecto, razón por la que requiere herramientas tecnológicas que le permitan ejecutar este monitoreo y el análisis de estos indicadores de una manera eficaz y viable, así como para replicar la experiencia a otros beneficiarios, acciones que buscan responder a la pregunta, ¿Cómo mejorar la gestión y monitoreo de lectura de los becarios de la Fundación Leonidas Ortega Moreira?

Pregunta problemática

¿Hasta qué punto el desarrollo de un sistema de gestión y monitoreo de lectura es una solución en la parte administrativa de la Fundación Leonidas Ortega?

Límite temporal de investigación

Un sistema de gestión y monitoreo de lectura es una opción factible para la administración de libros que se manejan a diario en la Fundación Leonidas Ortega. El problema nace por la necesidad de realizar un control exhaustivo de los lectores que adquieren libros y le dan uso, con el objetivo de toma de decisiones por parte de la Fundación Leonidas Ortega.

Límites espaciales de la investigación

Guayas, Guayaquil. Av. Vicente Rocafuerte Bejarano 520.

Objetivo general

Desarrollar un sistema de gestión y monitoreo de lectura, a través de un sistema web y aplicación móvil, para la gestión del proyecto de promoción de la lectura de la Fundación Leonidas Ortega.

Objetivos específicos

- Identificar las necesidades y requerimientos informáticos para la gestión y monitoreo de lectura de la Fundación Leonidas Ortega.
- Establecer las metodologías a utilizar, en el desarrollo del software informático y demás parámetros de arquitectura y patrones.
- Diseñar un sistema web y aplicación móvil que brinde las facilidades para gestionar y monitorear la lectura de los beneficiarios de la Fundación.
- Evaluar el sistema informático con pruebas funcionales realizadas con el usuario.

Variables

Según la fase de levantamiento de solicitudes y revisión de las ideas, se puede concluir que la variable independiente es el sistema informático, mientras que la variable dependiente es la gestión y monitoreo de lectura.

Sistema de gestión y monitoreo de lectura.

Sistema tecnológico que consiste en la gestión y monitoreo de lectura, llevando distintos módulos los cuales registran las actividades de los lectores con sus respectivos libros con relación a la Fundación Leonidas Ortega.

Justificación

"En el Ecuador se lee en promedio un libro completo y dos libros incompletos al año" (OEI, 2022) según estudios sobre los hábitos lectores, prácticas y consumos culturales en la población ecuatoriana, realizados por el Ministerio de Cultura y Patrimonio en coordinación la Organización de Estados iberoamericanos, el Instituto Nacional de Estadísticas y Censos y la Facultad Latinoamericana de Ciencias Sociales. En este contexto la Fundación Leonidas Ortega Moreira busca contribuir desde el proyecto de animación a la Lectura a motivar en sus estudiantes becados, el hábito lector como una herramienta de aprendizaje no sólo en contextos escolares, sino como una herramienta de aprendizaje para toda la vida, porque como diría José Antonio Millán "la lectura es la llave del conocimiento en la sociedad de la información" (Millán, 2008). Sin embargo, la implementación del proyecto requiere registros y controles que no serían factibles o eficientes llevarlos de manera manual, por lo tanto el contar con herramientas tecnológicas sin duda potencia las posibilidades de una ejecución exitosa del proyecto, por lo tanto el trabajo a desarrollar en esta tesis, busca brindar a la Fundación mejorar considerables para implementar su proyecto, al contar con un sistema web para el inventario y control de los libros donados a la Fundación para este proyecto y a través de la aplicación móvil para los estudiantes, la posibilidad de registrar los libros seleccionados para su lectura, monitorear el tiempo dedicado a esta actividad, así como su capacidad de análisis y síntesis.

Alcance de la Investigación

Los tipos de investigación seleccionados para el presente trabajo son investigación aplicada, exploratoria y descriptiva.

Investigación explicativa: Se aplican los conocimientos técnicos para el desarrollo de un sistema informático para la gestión y monitoreo de lectura mediante aplicación web y móvil.

Investigación exploratoria: Se brinda una solución informática para la Fundación Leonidas Ortega, siendo este un paso fundamental para abrir campos organizacionales hacia procesos tecnológicos, explorando procesos en los cuáles se encuentran involucrados la Fundación.

Investigación descriptiva: Se aplica el levantamiento de información y requerimientos de software, siendo este un objetivo que se debe cumplir en el ciclo de desarrollo de software. No obstante, se observan las metodologías, arquitecturas y tecnologías que se emplean durante el desarrollo del sistema informático.

Novedad o aspecto innovador

Según lo dicho por (Vásquez, Paredes, & Pérez, 2021) “La innovación es el centro de los planes de negocios presentes, las empresas son cada vez más generales y competitivas, exigiendo a los negocios a crear innovadores productos y servicios.” Las empresas u organizaciones que no innovan son las que pierden reconocimiento en el mercado o viabilidad como marca. Con el pasar de los años muchas organizaciones innovan sus servicios o productos junto con las tecnologías de la información. Cabe destacar que no es novedad que las empresas busquen nuevas herramientas tecnológicas para crecer organizacionalmente.

La innovación en el campo de la tecnología es recurrente debido a que es cambiante con el pasar del tiempo, cabe destacar que la Fundación Leonidas Ortega, opta por un sistema de innovación en el ámbito de monitoreo y gestión de los libros que maneja la organización.

Se da esto debido a la optimización de procesos y administrables, con el objetivo de mejorar las tomas de decisiones y controlar los estudiantes las cuales usan a diario libros, sin embargo, siempre es bueno es poner en práctica el conocimiento sobre la lectura realizada por el estudiante, de tal forma que este deba realizar una prueba tipo comentario para examinar los conocimientos adquiridos por el libro.

1 Capítulo I: Marco Teórico

El desarrollo de un sistema informático involucra muchas etapas y procesos que se deben cumplir de manera responsable, debido a que existen muchas variantes que hacen que un sistema tenga éxito, una de esas es el código limpio, buena arquitectura y metodología. Sin embargo, el éxito de un proyecto informático depende del valor humano y recursos que se brindan al proyecto. Según lo dicho por (Montoya, 2009) “Comprendemos por arquitectura en un proyecto de software a la capacidad conjunta y organizada de partes del sistema y del hardware con el propósito de cumplir una determinada acción”.

Es importante conocer las metodologías de desarrollo de software antes de iniciar, hay que tener bases sólidas sobre los métodos a aplicar en un sistema informático, de tal forma que se pueda escoger la mejor opción para cumplir con un desarrollo eficiente. No obstante, es importante señalar los principios sólidos y código limpio para un sistema mantenible con el tiempo. Las buenas prácticas en el desarrollo dan una buena satisfacción al usuario que usa el programa por ser un sistema eficiente que cumple con su objetivo y a la vez suele ser amigable, y para el desarrollador es fácil el entendimiento del código y modificaciones de forma mínimas y rápidas, por buenas prácticas usadas.

1.1 Metodología ágil de desarrollo SCRUM

Según lo dicho por (Pressman, 2010) los principios Scrum con el manifiesto ágil y se utilizan para guiar actividades de desarrollo dentro de un proceso de análisis que incorpora las siguientes actividades: requerimientos, análisis, diseño, evolución y entrega. Un importante término dentro de esta metodología es el llamado sprint, la cual define tareas con un rango vigente de fechas que debe cumplir el empleador.

Por lo general este flujo de trabajo es muy usado en empresas que cumplen con un cronograma de funcionalidades extensos. El personal de tecnología debe realizar reportes verbales o escritos de las actividades realizadas durante el día, llevando un control y gestión para la organización.

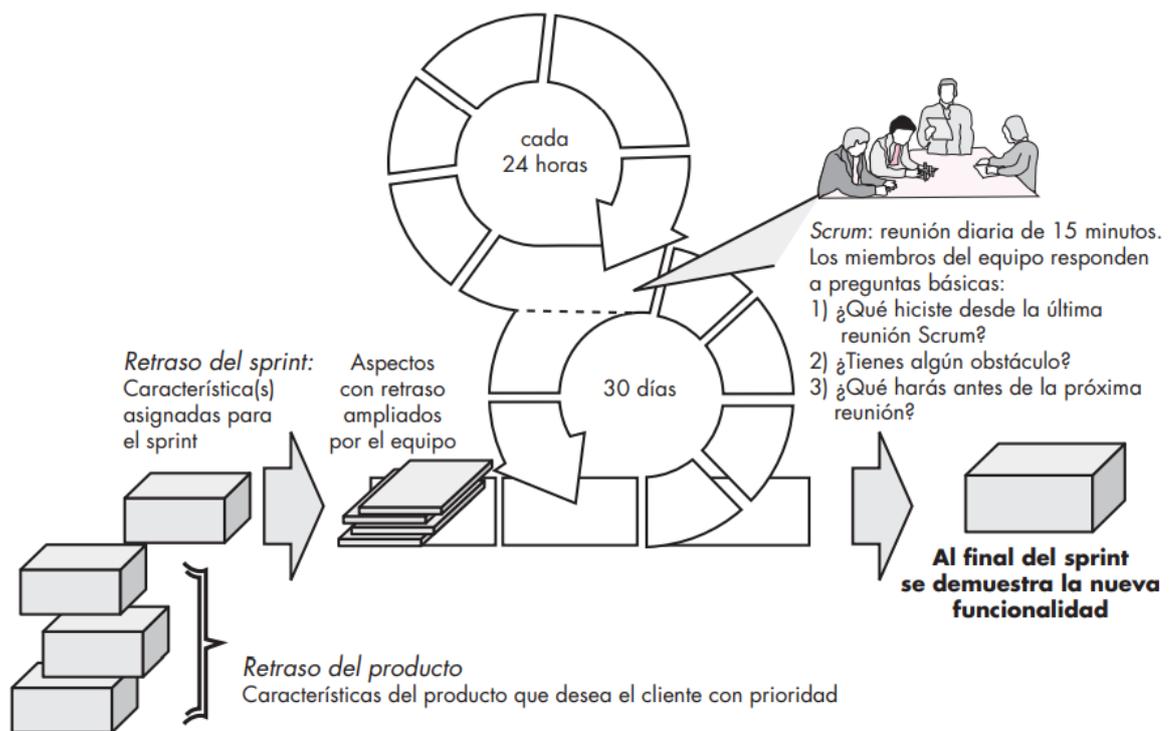


Ilustración 1: Modelo Scrum

1.2 Metodología en cascada

Según (Pressman, 2010) sugiere un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requerimientos por parte del cliente y avanza a través de planeación, modelado, construcción y despliegue.

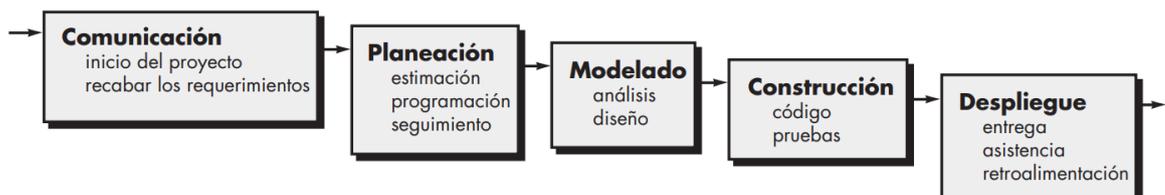


Ilustración 2: Modelo en Cascada

El modelo en cascada es el más antiguo de la ingeniería de software, con el pasar de los años este modelo ha presentado varios problemas, no obstante, ha sido eficaz para llevar a cabo sistemas informáticos. Una de las dificultades de este modelo es que no se lleva una secuencia del flujo que se realiza en el proceso. Aunque este modelo acepta repeticiones, lo hace en forma indirecta. Otra dificultad importante de este modelo es que los cambios funcionales no lo pueden verificar el cliente hasta que el proyecto esté muy avanzado.

1.3 Metodología evolutiva

Esta metodología está basada en los cambios constantes de los requerimientos cuando se avanza el desarrollo, esto hace que no se tenga una base sólida de lo que se desea desarrollar. No es muy usado en proyectos de gran alcance. Sin embargo, este modelo se puede caracterizar iterativo, debido a que se pueden desarrollar varias versiones del sistema. De este modelo han salido variantes tales como el modelo basado en prototipos y el espiral.

1.4 Modelo Espiral

Según lo dicho por (Olivera, Caridad, Alonso, & Manuel, 2020) El ciclo de desarrollo se representa como una espiral, en lugar de una serie de actividades sucesivas con retrospectiva de una actividad a otra. Lo mencionado hace referencia a que se desarrolla un sistema base o inicial con el objetivo de cumplir especificaciones abstractas. Esta metodología cumple con un ciclo de desarrollo que cuenta con cuatro fases las cuales son: definición de objetivos, evaluación y reducción de riesgos, desarrollo y validación, y planificación.

Este modelo cuenta cómo principales ventajas la buena comunicación con el cliente debido a que mezcla el modelo cascada y prototipo y los clientes pueden usar el sistema mientras se va desarrollando con cada incremento que se le da al software. Cabe destacar que esta tecnología es recomendada para proyectos con un gran alcance.

Sin embargo, hay que considerar que los incrementos se deben dar en cortos lapsos y considerando modulados por motivo de mitigar riesgos. Otro obstáculo

de esta metodología suele ser las dificultades para mantener planificaciones, y graficar el sistema globalmente.

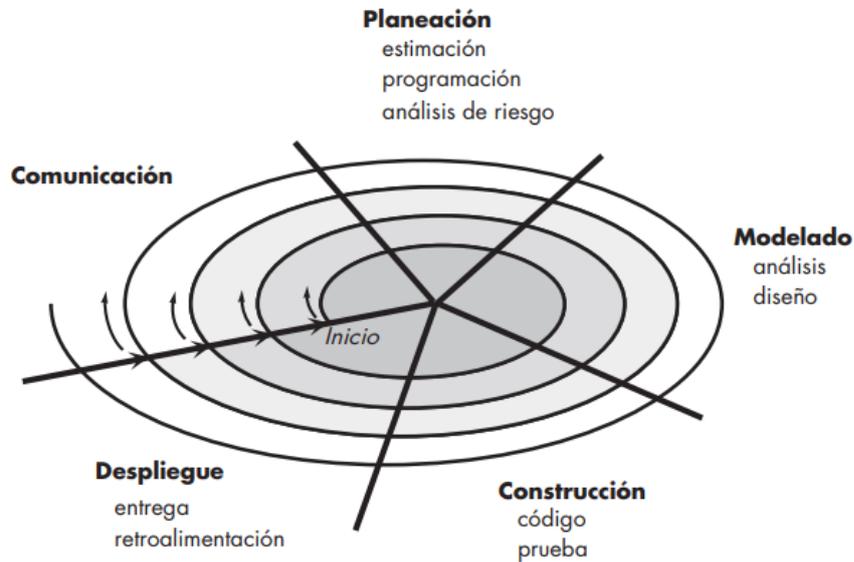


Ilustración 3: Ciclo de vida del modelo espiral. Fuente: (Pressman, 2010)

1.5 Modelo por prototipo

Según lo dicho por (Olivera, Caridad, Alonso, & Manuel, 2020) este modelo inicia con la recolección de requerimientos del cliente, con base a estos se define el conjunto de objetivos para el software, según eso se desarrolla de forma rápida un prototipo o maqueta que posteriormente la evalúa el cliente. El modelo de prototipo es muy usado cuando el software suele ser muy visual y de mucha interacción por parte del usuario, dando pautas de cómo es el sistema con respecto a colores, dimensiones, botones, entre otros. Para desarrollar sistemas hay que partir de una base de un diseño inicial.

Algunas de las ventajas de este modelo es la buena comunicación con los clientes y recomendado para proyectos pequeños y de medio alcance. Sin embargo, este modelo tiene desventajas tales como su planificación pueden sufrir cambios, periodos de desarrollo largo, respuesta a cambios en requisitos de usuario.

1.6 Programación orientada a objetos

La programación orientada a objetos es un paradigma muy frecuente en el desarrollo de sistemas informáticos. Este paradigma consiste en objetos, la cual nos permite construir sistemas complejos de una manera muy sencilla y organizada. Un objeto es un grupo consistido en clases y variables definidas. La etapa que define los objetos se denomina modelado de datos. La programación orientada objetos tiene terminologías aplicadas al desarrollo tales como:

Encapsulación: Uno de los principios de la programación orientada a objetos es la encapsulación. Cada objeto está comprometido con sus propios datos. Solo se puede cambiar a través de métodos propios del objeto. Por consiguiente, los atributos privados de un objeto no deben ser accesibles desde el exterior, por lo que los cambios solo se pueden realizar llamando a la función adecuada. Esto se logra protegiendo al estado del mal uso que podría tener consecuencias no deseadas.

Abstracción: La abstracción logra algo muy importante. Este es el poder de cambiar la implementación a voluntad. Desde fuera de la clase, los objetos que la usan no necesitan saber cómo se implementa internamente. Por ejemplo, si tiene una clase de repositorio responsable de devolver datos, debería poder cambiar la implementación sin que los objetos que usan ese repositorio lo sepan. Esto le da más flexibilidad.

Herencia: Esta fundamentado por la reutilización, por lo que puede definir relaciones jerárquicas entre clases y reutilizar atributos y métodos comunes.

Polimorfismo: Es la capacidad del objeto de la clase para proporcionar respuestas diferentes e independientes según los parámetros utilizados durante la llamada. Esto significa que los objetos como entidades pueden contener diferentes tipos de valores durante la ejecución del programa.

1.7 Patrón de diseño

Son plantillas que identifican problemas en el sistema y proporcionan soluciones previamente testeadas. Pueden relacionarse con la creación de objetos, las composiciones de clase, diseño y la comunicación entre objetos.

Los patrones de diseño sirven para validar nuestro código, pues representan estructuras previamente testeadas por un gran número de desarrolladores, con lo que nos ahorran una gran cantidad de tiempo y nos ayudan a ser más eficientes a la hora de llevar a cabo tareas y proyectos. Las categorías para clasificar los patrones son:

- Patrones creacionales: Relativos a la creación de objetos.
- Patrones estructurales: Correspondientes a las composiciones de clase y objetos.
- Patrones de comportamiento: Se refiere a la comunicación entre objeto de clase.

1.8 Arquitectura cliente servidor

La arquitectura cliente servidor es una de las arquitecturas más frecuentes y usadas durante el desarrollo de software. Por lo general, esta arquitectura suele consistir en un programa cliente y otro servidor donde se conectan entre sí, para la comunicación y transmisión de información continua. Cabe destacar que se suele establecer dos términos importantes los cuáles son back end y front end.

Back end: Es la rama o parte del desarrollo de software que involucra procesos relacionados a la lógica del negocio. Es decir, es la parte la cuál, el usuario final no ve, pero donde se controla y gestiona la información del negocio. Las tecnologías más comunes usadas en el desarrollo Back end son: Node JS, Nest JS, Laravel, Spring Boot, .Net Framework y Django.

Front End: Consiste en el apartado visual e interacción entre el programa y el usuario final. Es la capa o parte donde el usuario tiene mucho protagonismo de manera directa. Las tecnologías más comunes usadas en el desarrollo Front End son: Angular, React JS, Vue JS, Flutter, React Native y Ionic.

1.9 Tecnologías de desarrollo de software

Cuando se va a desarrollar un sistema informático se consideran muchos aspectos tales como metodología y arquitectura. Sin embargo, un aspecto fundamental es la elección de un stack tecnológico, empresas de renombre tales como Facebook, Amazon, Google tienen considerado las herramientas de desarrollo, y cabe destacar que estas empresas han creado herramientas tecnológicas. Con el pasar de los años las tecnologías suelen ser cambiantes y optimizadas, es importante conocer las nuevas tendencias y no morir con un sólo stack tecnológico, debido a que el respaldo por la empresa desarrolladora puede fracasar, y por consiguiente su tecnología, se considera conocer al menos dos tecnologías de las partes o ramas tecnológicas.

Al elegir un stack tecnológico se consideran los requerimientos del sistema. Según lo pedido por el usuario o cliente se desarrollan diagramas de procesos variados para que el desarrollador sepa que stack tecnológico puede usar. Ej: si los requerimientos del sistema consisten en el desarrollo de una aplicación móvil, el desarrollador debe elegir una tecnología front end para la construcción de interfaces gráficas interactivas para el usuario final. Además, seleccionar una tecnología de Back end para manejar el modelo de negocio y flujo de información. No obstante, se debe considerar un gestor de base de datos para la construcción del modelo entidad relación a emplear, y por último seleccionar una tecnología referente a despliegue de aplicaciones, puede ser servidores en la nube, los más conocidos y usados son: AWS, Azure, y Google Cloud.

1.10 Bases de datos relacionales

Según lo dicho por (Alejandro Manuel Rubinos Carvajal, 2011) se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. Las bases de datos relacionales son

estructuras de datos agrupadas relacionadas donde se almacena información. Los gestores de base de datos más comunes en el desarrollo de software son: PostgreSQL, MySQL, SQL Server y Oracle. Los gestores mencionados comparten algo más en común las cuales son las relaciones y normalización. Cabe destacar que cuando se desarrolla un sistema informático se elige un motor sea por simplicidad, fácil uso, seguridad y alta concurrencia. No bastante, para el presente proyecto se escogió SQL Server por su alta concurrencia y buena forma de diagramar entidades.

Nombre	Aplicado a:	Descripción:
SELECT	Tablas, vistas, secuencias.	Controla el derecho de seleccionar cualquier columna de una tabla o vista. También controla el derecho de interrogar a una secuencia.
INSERT	Tablas, vistas, secuencias.	Controla el derecho de insertar nuevos valores en una tabla, vista o secuencia.
UPDATE	Tablas, vistas, secuencias.	Controla el derecho de actualizar valores en una tabla, vista o secuencia.
DELETE	Tablas, vistas, secuencias.	Controla el derecho de eliminar valores en una tabla, vista o secuencia.
RULE	Tablas, vistas, secuencias.	Controla el derecho de crear nuevas reglas en una tabla o vista.
REFERENCES	Tablas.	Controla el derecho de relacionar dos tablas mediante una llave foránea.
TRIGGER	Tablas.	Controla el derecho de crear disparadores en una tabla.
CREATE	Bases de datos, espacios de tablas, esquemas.	Controla el derecho de crear nuevos esquemas dentro de una base de datos, nuevos objetos dentro de un esquema o nuevos índices (<i>o tablas</i>) dentro de un espacio de tablas.
TEMPORARY	Bases de datos.	Controla el derecho a crear tablas temporales dentro de una base de datos.
EXECUTE	Funciones.	Controla el derecho de ejecutar una función.
USAGE	Esquemas, lenguajes.	Controla la enumeración de objetos dentro de un esquema o el derecho a crear nuevas funciones con un determinado lenguaje de procedimiento.

Ilustración 4: Privilegios de una tabla. (Carvajal, Manuel, León, & Alina, 2011)

1.11 Autenticación de usuarios

Hoy en día los sistemas informáticos deben contener métodos de seguridad, sean a nivel de lógica de negocio, base de datos o protección de rutas de interfaces gráficas. Cabe destacar que la herramienta para proteger servicios REST más frecuente hoy en día es JSON WEB TOKEN (Auth0, n.d.) , está

herramienta proteger las llamadas APIS durante las consultas o peticiones que se realizan a nivel de servidor.

Los usuarios que se crean con esta herramienta se le asigna un TOKEN donde puede expirar días, horas y se puede realizar una renovación del mismo para que los usuarios externos no puedan hacer uso del API cuando un usuario exterior intente acceder. Los protocolos de seguridad se deben de manejar de forma adecuada y no dejar huecos de seguridad que involucren data sensible.

1.12 Selección de metodologías

Al elegir una metodología se analiza las diversas variantes por la cual se debe considerar una elección tales como tiempo de desarrollo, eficacia, ciclo de vida del software y mantenimiento. En el presente proyecto se consideró el uso de la metodología espiral basada en prototipos debido a que por motivos de pandemia de COVID 19 era complejo el uso de metodologías donde se presenta resultados diarios o reuniones frecuentes. Sin embargo, se opta por la metodología espiral debido a que su base no es definida por tiempos de entrega de resultados funcionales sino se basa en diseños de aplicaciones y cumple con ciclo de vida tradicional del desarrollo de software las cuales deben ser aprobados por el usuario final, dando como resultado el desarrollo de los procesos que están involucrados en la metodología.

La metodología espiral se suele usar en sistemas de medio alcance y se puede combinar con otras tecnologías, es muy frecuente mezclarlo con la metodología prototipos, ya que es la base principal, sin embargo, el presente proyecto fue basado en diseños concurrentes y presentados a la Fundación Leonidas Ortega. Lo destacable, es el rápido desarrollo por ser una metodología flexible y no estricta en tema de tiempos y resultados. Sin embargo, como toda metodología hay que cumplir un ciclo de desarrollo común la cual es: levantamiento de información, análisis de requerimientos, diseño, pruebas y mantenimiento.

2 Capítulo II: Marco Metodológico

2.1 Enfoque de investigación

Para este proyecto se utilizó un enfoque cualitativo, ya que la información fue recopilada por los involucrados de la Fundación Leonidas Ortega, sea personal que maneja el sector bibliotecario, entro otros. Al inicio del proyecto, se realizó una investigación exploratoria para explorar diversas soluciones a los problemas planteados por el área de enlace. Como resultado, hemos recopilado una gran cantidad de información e historias de usuarios que permitirán a los desarrolladores pasar a fases posteriores de requisitos. análisis. Se utilizó investigación descriptiva para construir un prototipo de validación y una interfaz basada en la entrada del usuario final. Finalmente, se utilizó el concepto de investigación aplicada para el desarrollo de software de control y monitoreo de lectura en conjunto con los conocimientos técnicos adquiridos en la carrera de ingeniero de sistemas.

2.2 Identificación y definición de variables

2.2.1 *Identificación de variables*

Variable independiente: La variable independiente del presente proyecto es el sistema informático, Por ende, es una variable causa donde ocurren variadas acciones involucrados en el sector bibliotecario de la Fundación Leonidas Ortega, como son: reportes y control de lectores involucrados.

Variable dependiente: Se determinó que la variable dependiente es la gestión y monitoreo de lectura debido que facilita un mayor control de las acciones envueltas en el sector bibliotecario de la Fundación Leonidas Ortega, y acciones que realizan los administradores o lectores.

2.2.2 Definición de variables

Variable independiente: Una variable independiente es una variable que se modifica o controla para estudiar su efecto. Actúa como condicionante de la variable dependiente. Se llama causal o experimental porque es manipulado por los investigadores.

Variable dependiente: Esta es la variable que se está investigando y midiendo. Variable afectada por la existencia o acción de un fenómeno o situación explicada, variable independiente.

2.2.3 Operacionalización de las variables

Tabla 1: Funcionalidad de las variables

Variable	Dimensión	Definición	Indicadores	Instrumento
Independiente = causa. Gestión y monitoreo de lectura	Administrador	Personal encargado del manejo de libros del sector bibliotecario de la Fundación.	Gestión de administradores.	Observación directa
	Lector	Usuarios que interactúan con los libros.	Gestión de lectores.	Observación directa
Dependiente = efecto. Sistema informático.	Estadísticas (web)	Estadísticas que se registran por cada interacción de préstamo de libro	Reporte de lectores y su interacción con libros.	Aplicación

	Comentario (web – móvil)	Registro de resúmenes del lector, y monitoreo por parte del administrador	Reporte de resúmenes e interacción directa con lector.	Aplicación
--	--------------------------	---	--	------------

2.3 Universo y muestra

2.3.1 Universo

Se ha definido como universo la Fundación Leonidas Ortega, en especial el sector bibliotecario, siendo este, el encargado principal del control y monitoreo de lectura de los estudiantes.

2.3.2 Muestra

Dado por la crisis actual del COVID 19 no es viable realizar el muestreo debido a que existieron restricciones que negaron el ingreso continuo a la Fundación Leonidas Ortega, especialmente presencia del desarrollador con el sector bibliotecario de forma presencial, por ende, se establece que la muestra es igual que el universo.

2.4 Métodos empíricos y estadísticos

Se realizó un proceso de entrevistas para recopilar información sobre el sistema a realizar, esto se dio por medio de la plataforma Google Meet y de forma presencial, obteniendo la primera lluvia de ideas, se consideró realizar prototipos para definir los modelos a seguir en el ciclo de desarrollo de software, según una base de información y prototipos. La gestión y monitoreo de lectura es de suma importancia para la Fundación Leonidas Ortega debido a que es bueno para este

ente conocer datos precisos sobre cuáles son los lectores involucrados con los libros o si en verdad están haciendo uso adecuado del material prestado.

La ilustración del modelo en espiral basado en el prototipo arrojó resultados efectivos para obtener retroalimentación del encargado de la biblioteca de la Fundación Leónidas Ortega y diseñar el sistema según lo requerido. Con el objetivo de potenciar el funcionamiento y el sistema de monitoreo y control de lectura, presentando diseños interactivos con la posibilidad de realizar reuniones para una posterior aprobación. Una foto de la invitación a la reunión se puede ver en el Anexo 2. Tenga en cuenta que el modelo de calidad del software GQM se utilizó para verificar el cumplimiento de los estándares técnicos, mientras que las pruebas de aceptación se verificaron mediante un documento llamado acta de pruebas. Por esta razón, se creó un manual de usuario que combina partes técnicas como la instalación y configuración para trabajar en modo desarrollo, e instrucciones para el usuario final que brindan detalles sobre cada pantalla, la facilidad de uso y las verificaciones realizadas en las mismas. El proyecto utiliza modelos para controlar la calidad y la mantenibilidad del software, en lugar de medir los procesos con gráficos estadísticos.

3 Capítulo III: Análisis e Interpretación De Los Resultados

Para el presente proyecto se realizó una entrevista principal para el respectivo levantamiento de información. Se conversó con personal involucrado en el manejo del software de la Fundación Leonidas Ortega. Adicionalmente, se crearon varias preguntas para darle al usuario una perspectiva por cual apunta el prototipo. Las preguntas fueron las siguientes.

- ¿Cuáles son las necesidades básicas?
- ¿Cuál es la imagen de la empresa en referencia a su marca, colores e ilustraciones?
- ¿Cuál es el alcance del proyecto?
- ¿Personas involucradas?
- ¿Qué recursos se dispone para el lanzamiento del software?
- ¿Módulos del sistema?

Las preguntas fueron respondidas en la primera reunión, las cuáles fueron de fundamental importancia para tener una retroalimentación y poder realizar los prototipos para la posterior aprobación del usuario. Se trabajó con la herramienta Figma para la elaboración de prototipos y facilitar la toma de decisiones en lo que respecta al modelado del sistema.

Cabe destacar que es importante especificar los puntos tratados en cada sesión, con la cual se definían los requerimientos, posteriormente, se describen las reuniones con las observaciones correspondientes.

Reunión 1

En la primera reunión que se tuvo con el personal de la Fundación Leonidas Ortega fue acogida de buena forma por ambas partes, compartiendo ideas y soluciones perspectivas. El modelo espiral aplicado fue de fundamental importancia para hacer un primer levantamiento de información dando como resultado lo detallado a continuación:

Véase la ilustración 17 del Anexo 2

- Listo de módulos: Estadísticas, libros y lectores.
- Comentario para lectores por libro.
- Login de usuario de administrador y lector.
- Libros visibles para el lector en app móvil.

Reunión 2:

En la segunda reunión se presentó el prototipo de las funcionalidades definidas. Para realizar las funcionalidades se hizo un previo análisis de requerimientos, se usó la herramienta Figma para prototipar el aplicativo móvil y web. No obstante, que antes de realizar el respectivo prototipo se realizó diagramas lineales, casos de uso y modelos de entidad relación, previo análisis sirvieron para hacer un buen prototipo que tenga como base los requerimientos definidos. En la segunda reunión se detalló lo siguiente:

Véase la ilustración 18 del Anexo 2 y la ilustración 10

- Seguridad de usuarios.
- Definición de datos a mostrar en estadísticas.
- Lista de libros no leídos y por leer para el lector.
- Pequeña sección de perfil de lector.

Reunión 3:

La reunión 3, se hizo por medio Google Meet, donde se revisaron los avances y se indicaron los detalles a realizar. Al obtener un sistema funcional, pero no terminado. Se indicó varias validaciones a seguir, a continuación, se detalla lo siguiente:

- Mínimo de palabras, en comentario.
- Correo único.
- Validaciones en inputs de datos (Textos, numéricos, alfanuméricos)
- Formularios submit valido.

Reunión 4

La reunión 4, se realizó una reunión presencial donde se presentaron los avances de las validaciones del sistema. Por consiguiente, se consideró las propuestas por parte del usuario para tener un sistema sólido y mejorar la experiencia de usuario. Cabe destacar, que en la presente reunión sirvió para validar la información propuestas con respecto a las validaciones de usuario.

Véase la ilustración 19, 21, 30 del Anexo 2

Reunión 5

En la presente reunión fue para presentar los avances del aplicativo web, dónde se visualizan las funcionalidades de la gestión de lectores, libros y Comentarios y la app Móvil donde se visualizan los libros prestamos al lector y la funcionalidad de realizar un comentario al libro seleccionado. El personal encargado de administrar el sistema de la Fundación Leonidas Ortega tuvo una aceptación debido a que cumple con lo definido en reuniones anteriores. El sistema fue intuitivo y funcional con las características elaboradas.

Véase la ilustración 20, 22, 23, 24, 25, 26, 27, 28 ,29 ,31, 32 del Anexo 2

4 Capítulo IV: Propuesta

En este capítulo se describe la implementación de la metodología escogida, se indican los términos y procesos que se involucran durante la ingeniería de software, incluyendo desde los principios de la metodología. Incluye arquitecturas, patrones, cronogramas, diseños, prototipos, etc. Con el objetivo de construir el sistema correctamente. También cubre conceptos clave a lo largo del ciclo de vida del desarrollo de software, también la metodología espiral mezclada con los principios del modelo prototipo.

4.1 Ingeniería de software

La ingeniería de software está basada por un conjunto de acciones, métodos y una lista de herramientas que habilita a los profesionales elaborar software de cómputo de alta calidad. (Pressman, 2010). Tenga en cuenta que la implementación de la ingeniería de software en el desarrollo de sistemas es rudimentaria, ya que las mejores prácticas deben considerarse en el punto del ciclo de vida del proyecto. Sin embargo, todos los desarrolladores deben conocer los principios básicos del ciclo del software. Esto se debe a que el sistema es mantenible, escalable, moderno, eficiente y, lo que es más importante, desempeña un papel clave para cumplir con las expectativas del usuario final.

4.2 Escenario actual

En la presente sección se muestra los casos que involucran el área de la biblioteca, administrada por los encargados del área. Por lo general, cuando se desea el préstamo de un libro, el lector se dirige directamente a la Fundación Leonidas Ortega, sector bibliotecario, y este le brinda un formulario de datos a llenar, proceso que se realiza de manera manual, y que puede provocar fallos por no controlar un excesivo manejo de los datos involucrados durante el préstamo del libro y acciones durante este, por ejemplo: duración del préstamo, comentario por parte del lector y estadísticas avanzadas que pueda proveer toma de decisiones. Cabe mencionar que un sistema de monitoreo, es una opción óptima para optimizar procesos que realiza manual la Fundación Leonidas Ortega. A continuación, se detalla los escenarios actuales.

Tabla 2: Situación actual de listado de lector. Fuente: elaboración propia (2022)

Escenario	Área de Biblioteca	
Sub escenario	Generar consulta de alumnos	
Actores	Jefa de Biblioteca	
Procedimiento	1.	Solicitud de identidad del lector
	2.	Verificación de datos del lector
	3.	Registro en listado interno de la biblioteca

Tabla 3: Situación actual de listado de libros. Fuente: elaboración propia (2022)

Escenario	Área de Biblioteca	
Sub escenario	Generar consulta de libros	
Actores	Jefa de Biblioteca	
Procedimiento	1.	Solicitud de ingreso de libro a la biblioteca
	2.	Verificación de datos del libro
	3.	Registro en listado interno de la biblioteca

4.3 Diagrama lineal de procesos actuales y del sistema

Un diagrama lineal de tareas muestra el orden en que se realizan las tareas en una organización y también muestra las relaciones lógicas entre todas las tareas que componen el diagrama. Las actividades realizadas en el desarrollo de software suelen representarse gráficamente en diagramas lineales o diagramas de flujo, que proporcionan una organización estructural para que el desarrollador de software comprenda claramente los procesos realizados en el sistema.

Diagrama lineal general sobre módulo de libro

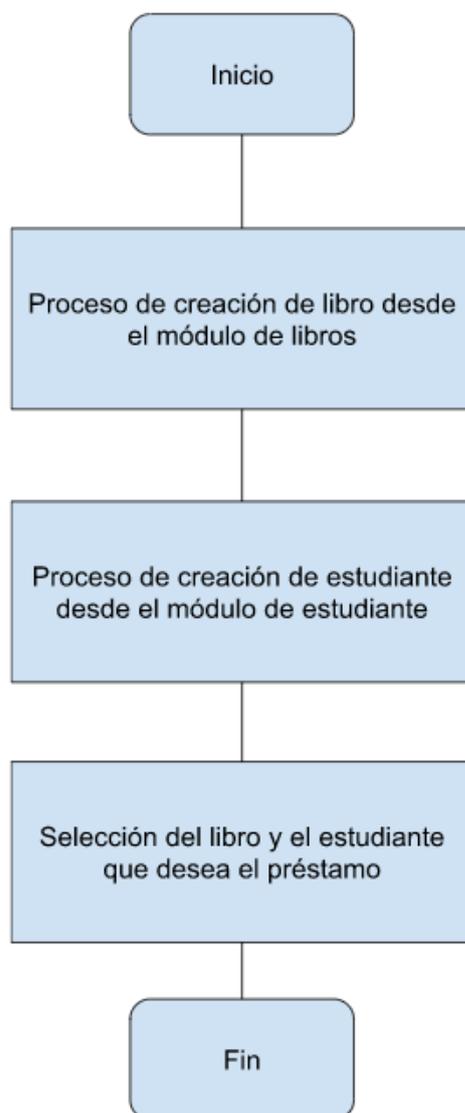


Ilustración 5: Diagrama lineal de préstamo de libros. Fuente: Elaboración propia (2022).

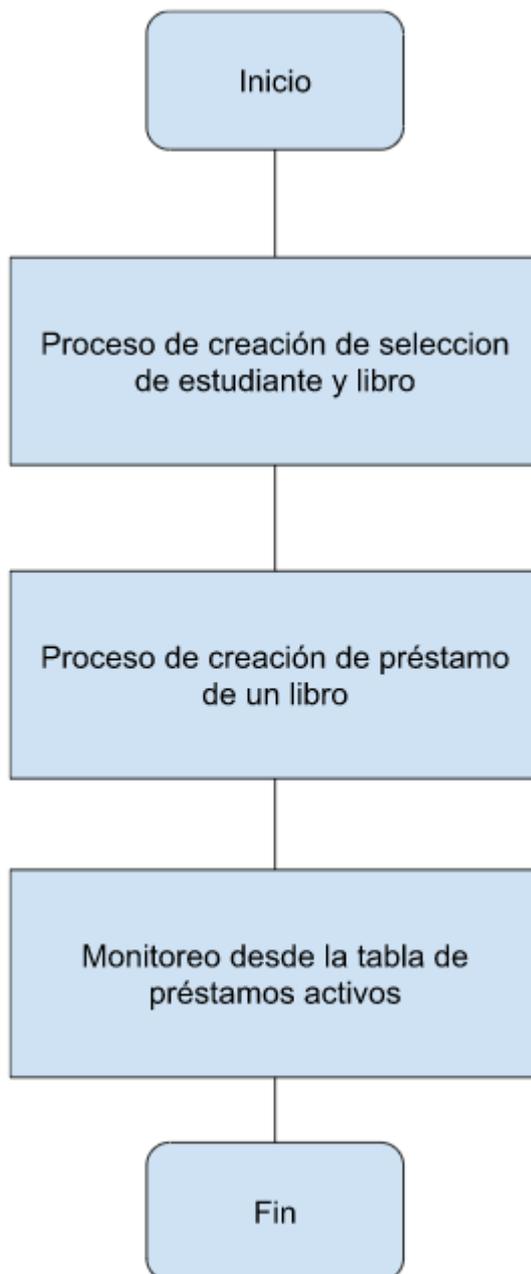


Ilustración 6: Diagrama lineal de Monitoreo de lectura. Fuente: Elaboración propia (2022).

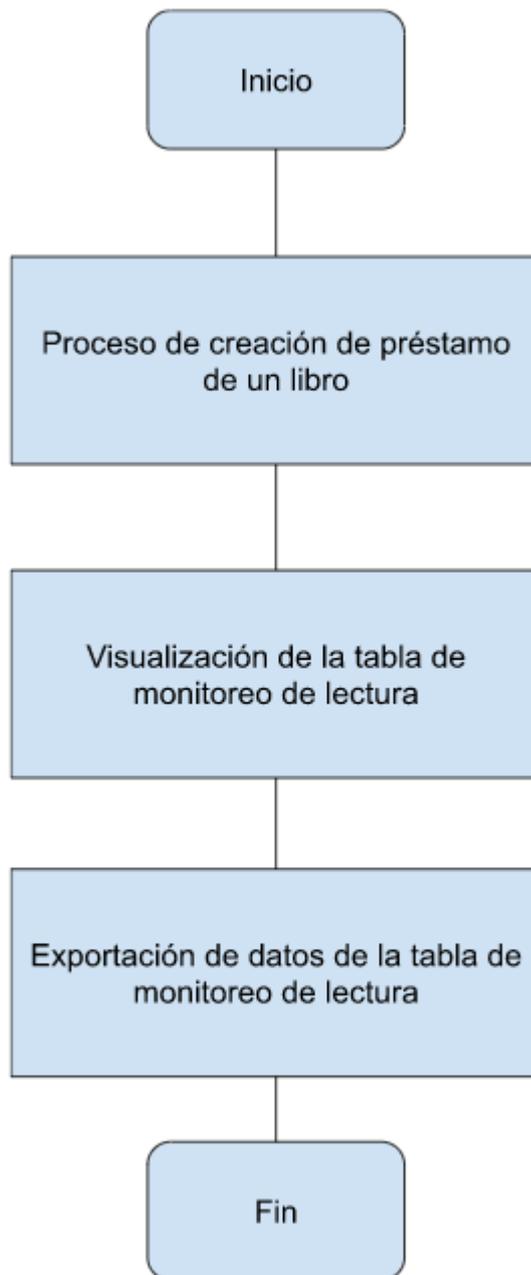


Ilustración 7: Diagrama lineal de Análisis de datos. Fuente: Elaboración propia (2022).

4.4 Aplicación de metodología

El modelo espiral es un modelo evolutivo utilizado para el desarrollo a largo plazo, con las mejores prácticas aplicadas a lo largo del ciclo de vida. Sin embargo, vale la pena señalar que la elección de este modelo se debe a varios factores externos que afectan a la parte humana. Esto se refiere al problema de la pandemia mundial de 2020 y utilizó metodologías ágiles como Scrum, XP y Kanban, suelen tener una estructura organizativa adecuada para reuniones constantes y análisis detallados durante el desarrollo de los sistemas informáticos, por lo que tenga en cuenta que la selección de modelos en espiral combinados con prototipos es muy actual en este proyecto. Todas las reuniones estuvieron abiertas a la presentación de prototipos de pantallas en línea, y en cada sesión se propusieron detalles.

4.5 Comunicación

4.5.1 Descripción del proyecto

El presente proyecto denominado desarrollo de un sistema informático para la gestión y monitoreo de lectura, mediante un sistema web y aplicación móvil, para la Fundación Leonidas Ortega nace ante la necesidad de optimizar procesos en la Fundación siendo la tecnología una herramienta de fundamental importancia para innovar en el sector bibliotecario. Cabe destacar, que el sistema tiene una automatización en el manejo total de control de lectores y libros. El sistema tiene como objetivo brindar soluciones a la Fundación dando un monitoreo y control de los préstamos de libro hacia el lector.

4.5.2 Alcance del proyecto

El proyecto cumple con los requerimientos funcionales y no funcionales posteriormente a especificar. Cabe destacar, que se usaron herramientas de desarrollo de software actual para brindar la mejor experiencia al usuario, siendo estas importantes para que el sistema sea mantenible. No obstante, se optó por arquitectura y metodología sostenible que ayude al software a ser un proyecto de gran escala.

4.5.3 Requerimientos

Un requisito de ingeniería de software es la necesidad de un sistema, programa o aplicación para satisfacer las necesidades y deseos del cliente. Los requisitos se dividen en partes de requisitos funcionales y no funcionales.

Requerimientos funcionales: Son solicitudes que tiene el software o aplicación y estas características interactúan directamente con el usuario. Los requisitos juegan un papel muy importante en el ciclo de vida del software. Los requisitos son datos que los desarrolladores consideran al construir sistemas informáticos. Comprender las actividades, los comportamientos y las funciones realizadas en relación con los requisitos de software funcional es fundamental. Los requisitos funcionales del software son los siguientes:

Tabla 4: Autenticación de acceso de administrador.

Detalle	Autenticación de acceso de administrador.
Prerrequisito	Cuenta de administrador
Entrada	correo y clave
Procedimiento	Validación de acceso de administrador.
Salida	Ingreso al sistema de gestión administrativa y muestra de módulos.

Tabla 5: Autenticación de acceso de lector.

Detalle	Autenticación de acceso de lector.
Prerrequisito	Cuenta de lector
Entrada	identificación y clave
Procedimiento	Validación de acceso de lector.
Salida	Ingreso a la app móvil y visualización de libros prestados

Tabla 6: Mantenimiento de listado de lectores.

Detalle	Visualización de listado de lectores.
Prerrequisito	Cuenta de Administrador
Entrada	lector
Procedimiento	Creación, edición y eliminación de usuarios lectores.
Salida	Visualización de lectores del sistema.

Tabla 7: Mantenimiento de listado de libros.

Detalle	Visualización de listado de libros.
Prerrequisito	Cuenta de Administrador
Entrada	libro
Procedimiento	Creación, edición y eliminación de libros.
Salida	Visualización de libros del sistema.

Tabla 8: Módulo de préstamos

Detalle	Módulo de préstamos.
Prerrequisito	Cuenta de Administrador
Entrada	Préstamo libro
Procedimiento	Creación, edición y cambio de estado de prestamos
Salida	Listado de préstamos realizados.

Tabla 9: Módulo de comentarios.

Detalle	Módulo de comentarios.
Prerrequisito	Cuenta de Administrador
Entrada	Lector

Procedimiento	Creación de comentario desde la App móvil
Salida	Visualización de comentarios en el administrador web

Tabla 10: Visualización de estadísticas

Detalle	Visualización de estadísticas.
Prerrequisito	Cuenta de administrador
Entrada	Libros, lectores y préstamos
Procedimiento	Creación de libros, lectores y préstamos
Salida	Estadística de la cantidad de libros, lectores y los distintos estados de los préstamos.

Tabla 11: Listado de préstamos solicitados

Detalle	Préstamos solicitados
Prerrequisito	Cuenta de Lector
Entrada	Préstamo libro por un administrador
Procedimiento	Creación de préstamo de libro desde el administrador web por un administrador e ingreso al perfil móvil del lector
Salida	Visualización de préstamos de libros solicitados

Requerimientos no funcionales: Estas son las solicitudes que explican lo que debe mantener un sistema para funcionar efectivamente. Validaciones y características que no son visibles para el usuario, pero esta es una forma de lograr que el sistema valide cada parámetro solicitado (por ejemplo, el número contiene 10 dígitos) para que el sistema no valide mensajes de error si se ejecuta. Estas acciones describen detalles para que el software sea lo más compatible posible en términos de verificación, carga, rendimiento, calidad del

software, etc. Los requisitos del sistema que no son de producción se describen a continuación.

Tabla 12: Requerimientos no funcionales

<p>Rendimiento</p>	<p>El sistema está construido sobre un modelo basado en componentes. En otras palabras, el sistema no tiene recargas de página, tiene componentes y tiene como objetivo tener un rendimiento de inicio de software de menos de 5 segundos.</p>
<p>Seguridad</p>	<p>El sistema usa una tecnología de seguridad llamada JWT que su objetivo es dar un token de autenticación al usuario, dando una protección a los EndPoints y datos del usuario.</p>
<p>Fiabilidad</p>	<p>El sistema tiene una fiabilidad del 100% porque contiene una protección de los datos a través de la tecnología de autenticación JWT, que habilita que tengan acceso solo los usuarios autenticados.</p>
<p>Disponibilidad</p>	<p>El sistema mantiene una disponibilidad del 100% del tiempo a la internet, debido a ser una aplicación web y móvil.</p>
<p>Portabilidad</p>	<p>El sistema mantiene una accesibilidad desde los siguientes navegadores web: Safari, Chrome, Edge,</p>

	Firefox, Opera y Microsoft, para dispositivo móvil: Android e IOS.
Usabilidad	El sistema tiene una interfaz sencilla y moderna que es fácil de entender por el usuario dado que este cuenta con un menú de los módulos respectivos del sistema, sobre todo con colores y logos representativos de la Fundación Leonidas Ortega.
Software	<ul style="list-style-type: none"> • Base de datos SQL SERVER • Angular v13 • Flutter • Typescript • Web Services con NEST JS - REST API

4.6 Planeación

4.6.1 Estimación de tiempos

La estimación del tiempo se refiere a las actividades de planificación dentro del ciclo de vida del software. El tiempo estimado para cada proceso involucrado es importante. De esta forma, se puede mejorar la estructura organizativa entre clientes y desarrolladores, ya que se tiene en cuenta el tiempo durante la fase de desarrollo.

4.6.2 Análisis de riesgo

El análisis de riesgos es importante en el modelo espiral debido a que se destaca como principal característica. Esto se hace con el objetivo de obtener un software libre de errores y efectivo según lo prescrito por las evaluaciones de prueba consideradas en prototipos constantes. A continuación, se muestra una matriz de posibles riesgos en el software.

4.6.3 Matriz de riesgos

Tabla 13: Matriz de riesgos. Fuente: propia (2022)

Riesgo	Impacto (1-5)	Probabilidad	Mitigación	Contingencia
Solicitudes funcionales no consideradas durante el levantamiento de información	4	75%	Comunicación con el cliente, construcción de prototipos.	Desarrollo del requerimiento solicitado por el cliente y el tiempo estimado.
Comprobaciones no aprobadas por el usuario	3	45%	Contemplar reunión para las correspondientes pruebas de los módulos del sistema.	Cumplir con pruebas durante la presentación de prototipos.
Ruta de API no protegida, con información interna	2	35%	Seguridad de rutas de API por medio de JWT	Durante la mantenibilidad del sistema realizar pruebas y protección de rutas del API
Validaciones de campos no contemplados	3	45%	Realizar las validaciones correspondientes y probadas	Realizar el mantenimiento adecuado, considerando las validaciones

			antes de pasar a producción	correspondientes.
Mantenimiento y nuevas funcionalidades	2	25%	Desarrollar el manual técnico correspondiente	Uso del manual para futuras funcionalidades en el sistema informático

Esta matriz de riesgos entrega obligaciones y posibilidades que pueden ocurrir en los proyectos de software. Vale la pena señalar que se han evaluado cinco posibles riesgos y sus soluciones. Dentro del modelo en espiral, las consideraciones de matriz son muy importantes. A diferencia de otros métodos, el modelo se caracteriza por una evaluación de los posibles riesgos.

4.7 Modelado

4.7.1 Análisis

Descripción del producto

Tabla 14: Descripción del producto

Inicio	Autenticación de usuario	Ingreso al sistema de información con credenciales de acceso
Administradores	Módulo de libro	Crear, editar y eliminar libros
	Módulo de lector.	Crear, editar y eliminar lector

	Módulo de comentario	Gestionar los comentarios realizados por el lector
	Estadísticas	Estadísticas sobre los libros consumidos por los lectores y tiempo de lectura.
Lector	Perfil de usuario	Visualización de datos personales del usuario
	Creación de comentario	Creación de comentario de libro seleccionado

Viabilidad del proyecto

Un estudio de factibilidad actual del proyecto permitirá evaluar si el sistema es óptimo para las condiciones de la Fundación Leonidas Ortega. La viabilidad se divide en tres partes.

Viabilidad operativa

Un sistema informático automatiza muchos de los procesos asociados al área de biblioteca de la Fundación Leónidas Ortega, permitiéndonos hacer un seguimiento de toda la actividad que se lleva a cabo en esta área en particular. Ejemplo: En el pasado, al prestar un libro, se creaba un Excel o documento para los registros del lector, y mediante un sistema informático, estos registros se almacenaban en un software de gestión sin pérdida del documento, lo que permitía referenciar datos adicionales. Fechas exactas de préstamo de libros, los administradores tienen la oportunidad de mantener informes sobre préstamos. Operacionalmente es viable debido a la automatización de los procesos que

realiza el sistema para la Fundación Leonidas Ortega, convirtiendo al software en una herramienta fundamental para el seguimiento y control.

Viabilidad técnica

Tabla 15: Viabilidad técnica

Análisis y Diseño del Sistema	Modelo de desarrollo tipo espiral con prototipos
Software de Desarrollo	Angular / Flutter / SQL server/ Nest JS
Lenguaje de Programación	Typescript y Dart
Base de Datos	SQL server
Documentación del Proyecto	Microsoft Office

La viabilidad técnica se refiere a la evaluación de componentes, herramientas y equipos para que el software de computadora funcione correctamente. Sin embargo, la viabilidad técnica permite el análisis de factores humanos, siempre que los factores humanos posean los conocimientos necesarios para desarrollar un sistema informático y permitir el mantenimiento posterior por parte de personal autorizado. Esta tabla muestra los requisitos de software.

Viabilidad económica

En el clima económico actual, los servidores en la nube son valorados como proveedores de servicios por los planes que manejan y la estabilidad que ofrecen. Uno de los beneficios de los servidores en la nube es que no tiene que proporcionar mantenimiento de equipos físicos o reemplazos de discos duros. Esto se debe a que estos tipos de servicios son administrados por una organización y dependen de los servidores de los usuarios para administrar, configurar e instalar componentes. Se considera que los componentes y recursos necesarios para implementar el sistema son cubiertos por la Fundación Leonidas Ortega.

Conclusiones de estudio de Viabilidad

Con base en la investigación realizada y los parámetros estimados, se cree factible el desarrollo de un sistema informático para el seguimiento y control de la Fundación Leonidas Ortega, el cual se detalla en los siguientes puntos.

- El software es una herramienta que responde a las necesidades locales y brinda mucho apoyo a los trabajadores de la Fundación Leonidas Ortega.
- Habilidades adquiridas para desarrollar con éxito software.
- El autor lleva a cabo un curso de formación para el personal de la biblioteca.
- Los costos de implementación del software son cubiertos por la Fundación Leonidas Ortega.

4.8 Diseño

4.8.1 Prototipos

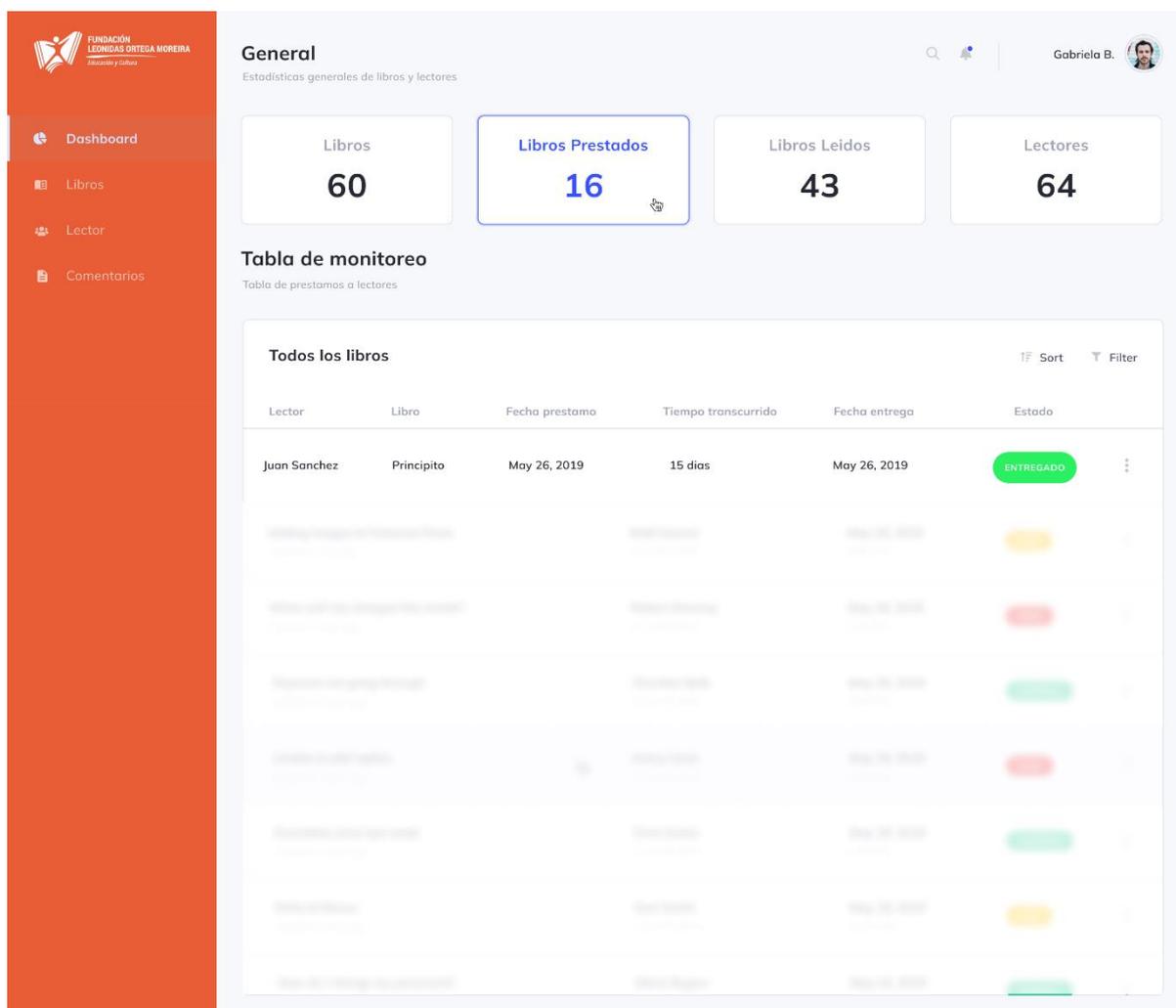


Ilustración 8: Administrador web. Fuente: propia (2022).

La presente ilustración representa el administrador web del sistema, donde se visualizan los módulos del sistema, las cuáles son Estadísticas, libros, lector y comentarios. La imagen de marca se caracteriza por los colores y tipografía de la Fundación Leonidas Ortega. El presente prototipo fue presentado en la segunda reunión con el personal respectivo. Según el levantamiento de información y realizado un análisis se dedujo la elaboración del prototipo presente para los usuarios administradores.

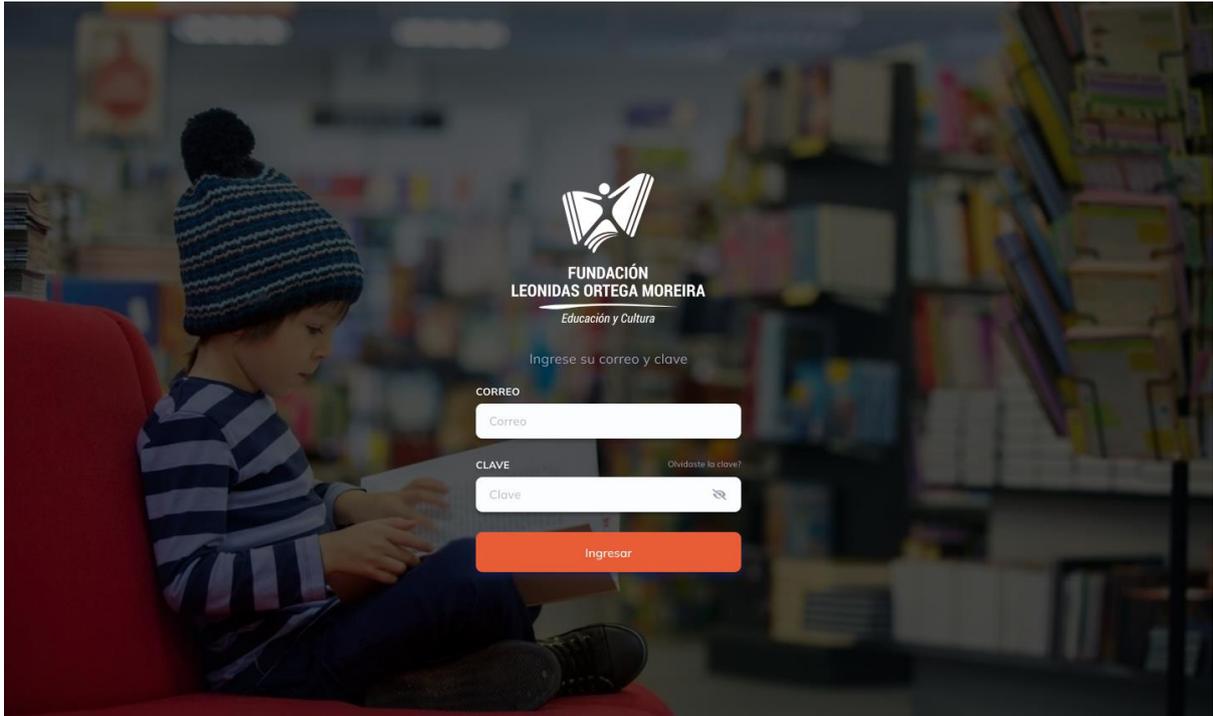


Ilustración 9 Prototipo de Login. Fuente: Propia (2022)



Una vez conversado con el personal de la Fundación sobre temas de seguridad en el aplicativo web y Móvil, se desarrolló el prototipo que los usuarios de la Fundación Leonidas Ortega interactúen, un diseño representativo a educación, donde se visualiza únicamente 2 campos que son correo y clave. A nivel de código se implementa medidas de seguridad a detallar en los próximos puntos al referirse de construcción de código.

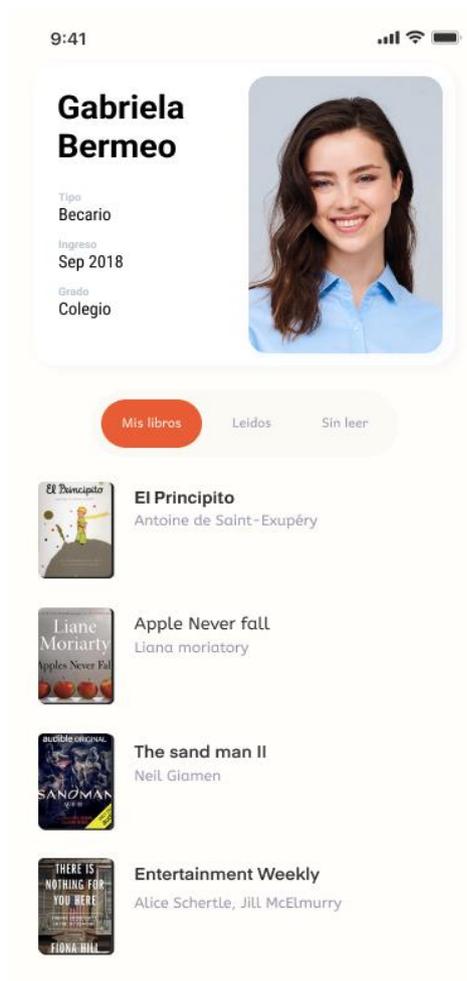


Ilustración 10: Perfil Móvil Fuente: Propia (2022)

En la presente ilustración se visualiza el perfil de lector junto con sus segmentos de libros del lector, clasificados en leídos y sin leer. Dado los requerimientos y prototipos entregados en la tercera reunión con el personal de

la Fundación Leonidas Ortega se pudo presentar el diseño del aplicativo móvil junto con sus características, la cual fue del agrado del personal.

4.8.2 Diagrama de casos de uso

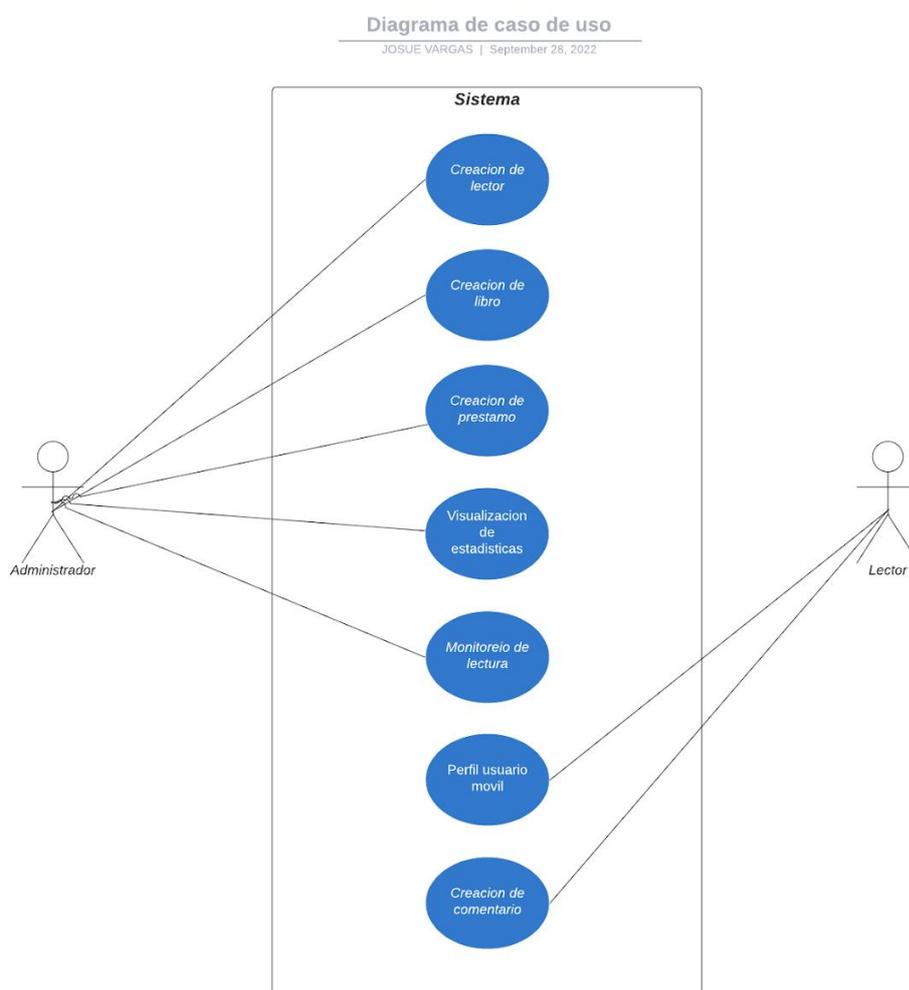


Ilustración 11: Diagrama de casos de uso

4.8.3 Arquitectura basada en componentes y Widgets

La ingeniería de software fundamentado en componentes es un proceso centrado en el diseño y la construcción de sistemas basados en computadora que utilizan "componentes" de software reutilizables (Pressman, 2010). Por qué este software utiliza un modelo basado en componentes Se eligió un software muy popular para construir interfaces gráficas para el desarrollo de sistemas informáticos. Esta herramienta tecnológica es Angular, desarrollado y mantenido por Google. Esta herramienta ayuda a los desarrolladores a crear sistemas mediante la implementación de las mejores prácticas de programación, como: Orden de carpetas y archivos. Angular utiliza el concepto de componentes. Es básicamente una parte reutilizable para el desarrollo y uso en múltiples pantallas diferentes o hacia un objetivo común. Este concepto se ve reforzado por las potentes funciones de programación asíncrona y reactiva de Angular.

La arquitectura Angular implementa los conceptos de aprendizaje necesarios para usar este framework correctamente. Angular CLI es una herramienta que combina bien con el marco y lo ayuda a crear nuevos módulos, servicios, páginas y componentes mediante la inyección de dependencia. Trabajar con Cli es una gran ventaja al hacer proyectos en Angular, brindando beneficios que hacen que el desarrollo sea eficiente, rápido y moderno, que siempre se basa en la aplicación de las mejores prácticas arquitectónicas y la programación reactiva.

En el mundo de las aplicaciones móviles, existen diversas tecnologías o soluciones para el desarrollo de apps. Cabe destacar que Flutter es una tecnología creada y mantenida por Google, esta tecnología provee una gran cantidad de herramientas que permiten desarrollar el Layout de una forma muy cómoda, los desarrolladores no deben ignorar esta herramienta debido a que el crecimiento se da por el uso multiplataforma y gratuito, a partir de una sola base de código.

El toolkit de flutter está diseñado para poder crear apps vistosas y muy rápidas sin poder afectar el rendimiento, algo que muchas tecnologías prometen pero que no las cumplen del todo. El lenguaje primordial de Flutter es Dart, que

utiliza muchas características de los lenguajes modernos, como reactividad, promesas y observables. Flutter compila a código de máquina, y su árbol de widgets está basado en una serie de piezas o componentes conectados.

4.8.4 Arquitectura cliente servidor

Esta arquitectura fue elegida por su robustez al trabajar con aplicaciones web y, lo más importante, la separación de la lógica comercial y los datos presentados al usuario. La arquitectura divide el software en dos partes:

Cliente

Esta sección representa la parte visual que se muestra al usuario. Esto incluye la programación de interacciones con datos, la elaboración de prototipos e interfaces gráficas. Los clientes tiene un rol muy importante en el desarrollo de los sistemas informáticos, dependen de la interacción del usuario con el software, la optimización de recursos y las demandas constantes de aplicaciones como servidores.

Servidor

Esta parte describe la creación de información en aplicaciones similares a un servidor denominadas servicios web. El objetivo de esta aplicación es construir una API REST como arquitectura y ejecutar consultas en el cliente a través del protocolo HTTP. La configuración e implementación de la arquitectura de carpetas se especifica en el servicio web para que funcione bien con la API REST. La lógica implementada está en un archivo llamado Controladores que está completamente vinculado a responder a la base de datos SQL SERVER utilizada por el proyecto en cuestión, y otro archivo llamado Rutas que contiene las URL solicitadas a través de HTTP.

4.8.5 Diccionario de datos

A continuación, se encuentran las definiciones y descripciones de los datos utilizados en el sistema de gestión. Primero, nombra cada entidad con su descripción y la tabla que contiene los campos y sus detalles.

Libro					
ATRIBUTO	TIPO DE DATO	TAMANO	DEFAULT	PK	AUTOINC.
ID	INT	10		X	X
NOMBRE	VARCHAR	50			
AUTOR	VARCHAR	50			
RESUMEN	VARCHAR	100			
FOTOPORTADA	VARCHAR	100			
ESTADO	VARCHAR	1	A		
FECHACREACION	DATETIME	8			
FECHAMODIFICACION	DATETIME	8			

LibroLector					
ATRIBUTO	TIPO DE DATO	TAMANO	DEFAULT	PK	AUTOINC.
ID	INT	10		X	X
IDLIBRO	INT	10			
IDLECTOR	INT	10			
ESTADO	VARCHAR	1	A		
FECHACREACION	DATETIME	8			
FECHAMODIFICACION	DATETIME	8			

Lector					
ATRIBUTO	TIPO DE DATO	TAMANO	DEFAULT	PK	AUTOINC.
ID	INT	10		X	X
NOMBRES	VARCHAR	40			
APELLIDOS	VARCHAR	40			
EDAD	INT	10			
CORREO	VARCHAR	40			
GRADO	VARCHAR	40			
FOTOPERFIL	VARCHAR	40			
ESTADO	VARCHAR	1	A		
FECHACREACION	DATETIME	8			
CLAVE	VARCHAR	50			
FECHAMODIFICACION	DATETIME	8			
IDENTIFICACION	VARCHAR	11			
INSTITUCION	VARCHAR	100			

Administrador					
ATRIBUTO	TIPO DE DATO	TAMANO	DEFAULT	PK	AUTOINC.
ID	INT	10		X	X
NOMBRE	VARCHAR	40			
CORREO	VARCHAR	40			
CLAVE	VARCHAR	40			

Comentario					
ATRIBUTO	TIPO DE DATO	TAMANO	DEFAULT	PK	AUTOINC.
ID	INT	10		X	X
DESCRIPCION	VARCHAR	200			
IDLIBROLECTOR	INT	10			
FECHACREACION	DATETIME	8			
FECHAMODIFICACION	DATETIME	8			

4.8.6 Modelo entidad relación

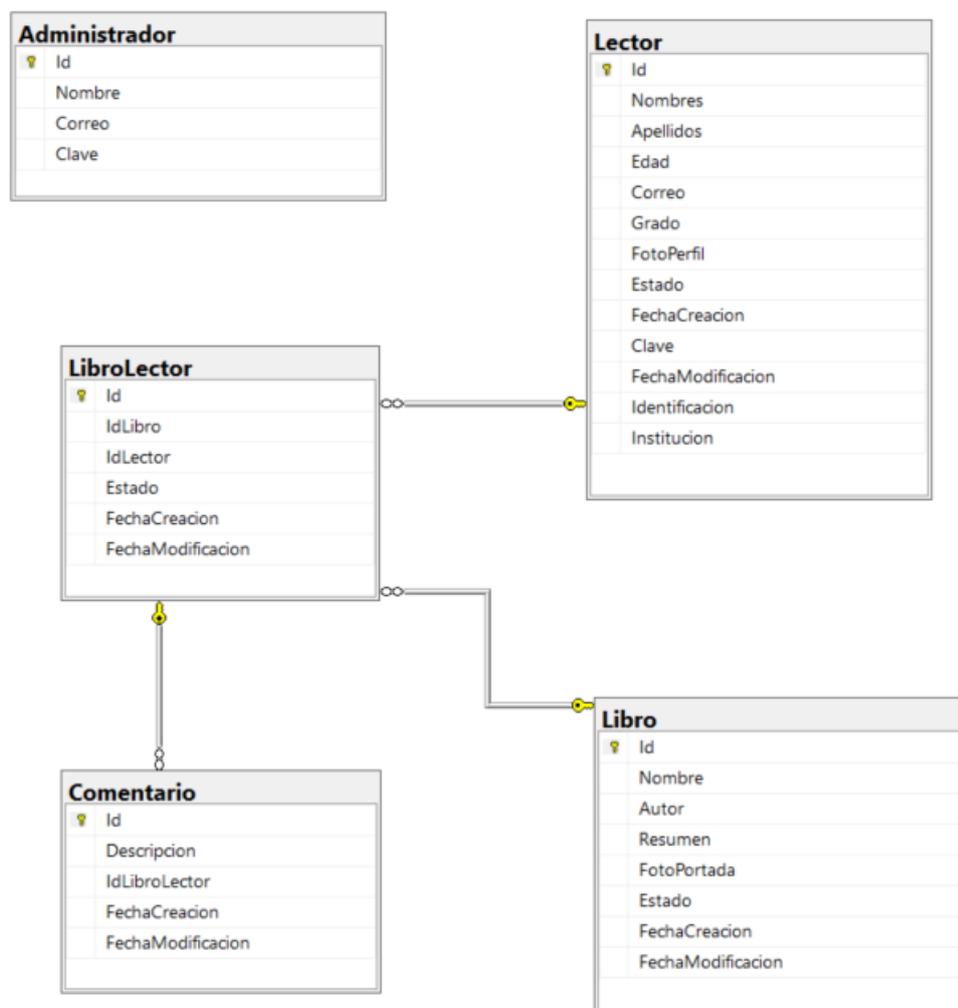


Ilustración 12: Modelo de base de datos

4.9 Construcción

4.9.1 Código

Después de que se realiza cada análisis de la información y se diseñan las tablas y la estructura del programa, se considera el siguiente paso llamado codificación. En esta fase, el desarrollo del programa comienza con un análisis y desarrollo de la lógica en un editor de código. La codificación se realiza según la arquitectura elegida (en este caso cliente-servidor). El código está escrito en el servidor en Web Services, el lenguaje utilizado para desarrollar con Nest JS.

La arquitectura REST API se utiliza en los servicios web, que son muy utilizados en este tipo de servicios. Con esta arquitectura, el desarrollador responsable de desarrollar la aplicación cliente siempre realizará solicitudes al servicio y el servicio responderá en consecuencia. La configuración de los datos es muy importante en las aplicaciones de servicios web y depende de ello para la información que se presenta al usuario. Sin embargo, es relevante tomar en consideración la seguridad del servicio. Esto garantiza que la ruta esté protegida de acuerdo con el protocolo HTTP y disponible para cualquier cliente válido del sistema (referido como autenticado en este proyecto). El código para el servicio web se describe a continuación.

```
lector.controller.ts x
src > lector > controllers > lector.controller.ts > ...
14     }
15
16     @Get()
17     findAll() {
18         return this.lectorService.findAll();
19     }
20
21     @Get('Count')
22     countAll() {
23         return this.lectorService.countAll();
24     }
25
26     @Get('/:id')
27     findOne(@Param('id') id: string) {
28         return this.lectorService.findOne(+id);
29     }
30
31     @Put('/:id')
32     update(@Param('id') id: number, @Body() updateLectorDto: UpdateLectorDto) {
33         return this.lectorService.update(+id, updateLectorDto);
34     }
35
36     @Delete('/:id')
37     remove(@Param('id') id: string) {
38         return this.lectorService.remove(+id);
39     }
40 }
```

Ilustración 13: Controlador de lector. Fuente: propia (2022)

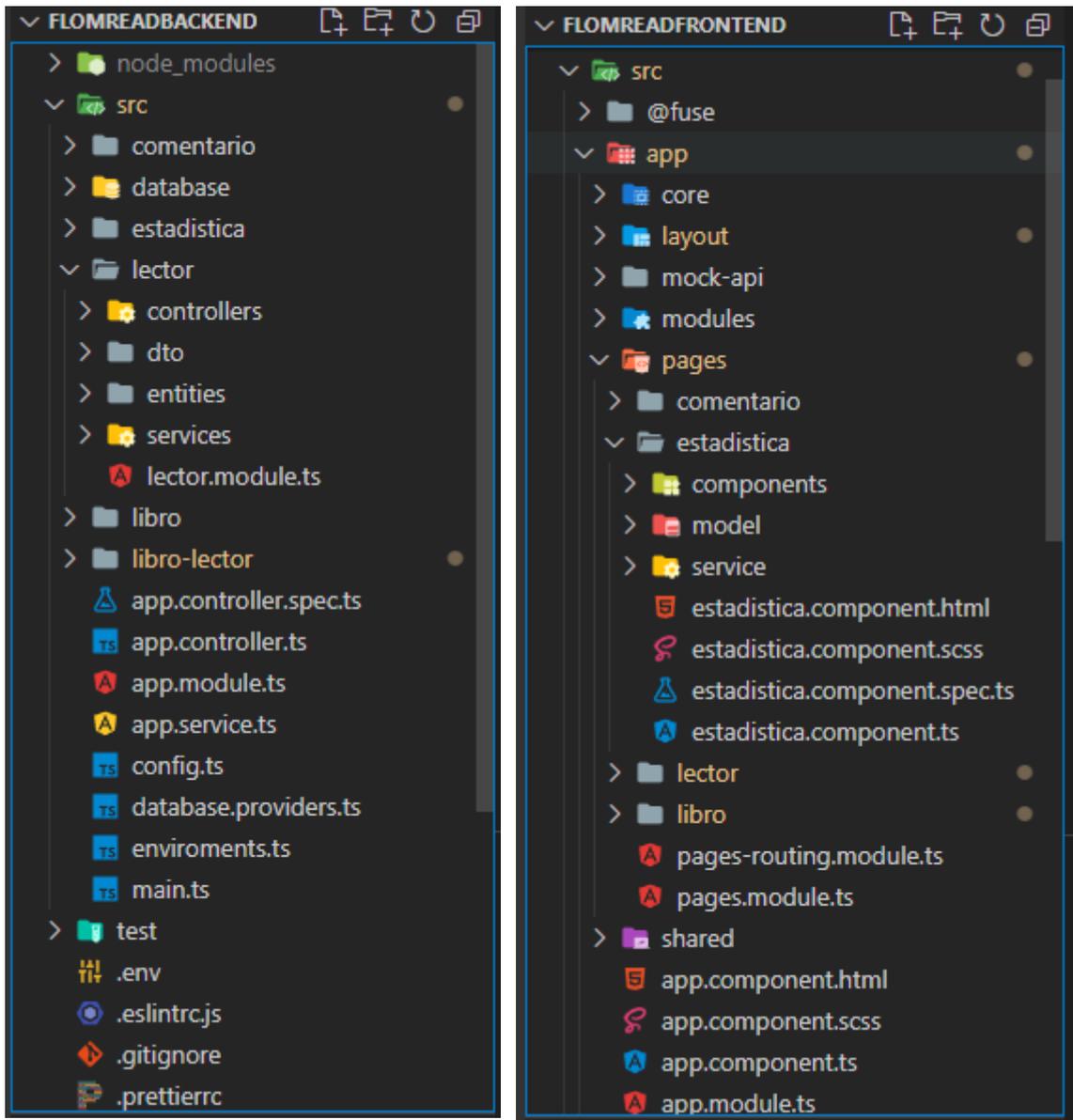


Ilustración 14: Estructura de carpeta de Nest js y Angular

4.9.2 Prueba

Las pruebas de software son, con mucho, la función de control de calidad más común que se realiza en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas. (Sanz, 2005). Actualmente existen una serie de

estándares de evaluación de software, incluidos los estándares ISO y otras empresas que realizan la certificación de software. La lista de estándares internacionales ISO/IEC 1550 es una referencia para evaluar avances del sistema informático(ISO/IEC, 2003). (Dávila, García, & Córdor, 2016) Más adelante se explicarán en detalle los tipos de pruebas disponibles y su elección para este proyecto.

Pruebas de aceptación

Las pruebas de aceptación son pruebas formales relacionadas con las necesidades del usuario, los requisitos y los procesos comerciales. Se utilizan para determinar si un sistema cumple con los criterios de aceptación que permiten a los usuarios, clientes u otros organismos autorizados decidir si aceptan el sistema. Estas pruebas se desarrollan en base a casos de uso, procesos de negocio e informes de análisis de riesgo y se utilizan para validar los procesos de negocio del sistema. Procedimientos de usuario, formularios, informes y datos de configuración. Con base en el primer día de pruebas de usuarios, las pruebas de integración actuales consideraron los informes de prueba con los usuarios finales.

Modelo de calidad de software GQM

Este es un enfoque que proporciona una manera conveniente de definir métricas de proceso y resultado para su proyecto. Este modelo de calidad se utilizó debido a la necesidad de desarrollar conjuntamente objetivos conjuntos relacionados con la productividad y la medición de la calidad. Además de generar preguntas que permitan establecer metas de manera completa y cuantitativa. Además, el desarrollo de acciones relacionadas con el desarrollo de productos y procesos relacionados con los objetivos.

La relación con la aplicación de gestión de asistencia, ya sea desarrollo de automatización de procesos, mide cuánto tiempo se dedica al código, cuántas clases se implementaron ese día, cuántos errores se resolvieron, etc. Se trata

de medir métricas. Este modelo de calidad es integral para ingresar preguntas y respuestas específicas de terceros con respecto a la mejora del software.

Seguridad en las comunicaciones

El servidor y el acceso del cliente está dada por la arquitectura cliente-servidor y el cliente utilizado es Angular. Ejecuta la solicitud que se muestra en la consola de ejecución de la herramienta en el servidor. Puede solicitar json desde el servidor a través de HttpClientModule proporcionado por Angular.

Formato de archivo

La respuesta del servidor se devuelve en formato JSON. Esto se debe a que la gestión y manipulación de datos es más fácil de desarrollar. Sin embargo, el formato JSON es un formato muy común en las arquitecturas cliente-servidor.

Requisitos funcionales de Modelo de Calidad

- **Gestión de proyecto:** El software de Control de Gestión cuenta con controles de seguridad relacionados con sus reglas de privacidad proporcionados por la plataforma tecnológica JWT.
- **Accesibilidad y usabilidad:** el sistema controla el acceso y solo permite a los usuarios autorizados a través de una propiedad de autenticación que contiene un token interno proporcionado por el backend.
- **Búsqueda de información:** un servicio de canal de búsqueda que ayuda a crear una lista limpia de lo que los usuarios han escrito en sus búsquedas, proporcionando detalles de lo que necesitan y predicciones similares.
- **Documentación generada:** el software incluye estándares de mejores prácticas, lo que da como resultado un código limpio, bien documentado y fácil de entender para el mantenimiento del sistema. Los programadores están más organizados debido a la arquitectura MVVM, que estandariza la organización del software a través de la arquitectura del software.

Estándares y guías:

Objetivos del software según modelo GQM - Desarrollo de un sistema de control de gestión informatizado del área conectada que cumpla con los estándares de calidad.

Propósito de las mediciones:

Se realizan pruebas funcionales para los evaluadores de cada módulo para el diseño, la consulta y la carga de componentes, y la modificación de las opiniones de software relacionadas con la exportación de informes.

Proceso:

Pasos durante la fase de desarrollo con prototipos para proporcionar retroalimentación y permitir el desarrollo continuo. Se supone que cada módulo se completó para avanzar en la fase de prueba.

Recursos:

El framework Angular crea archivos para pruebas unitarias de componentes, estos archivos permiten poder verificar si se llaman de forma efectiva la inyección de independencia, solicitud independiente de las acciones involucradas en la clase y propiedades no usados.

Para la valoración respectiva del sistema se expuso preguntas para valoraciones.

De acuerdo al Nivel Operacional:

¿Los módulos del software de monitoreo y control son los adecuados y van acorde al objetivo planteado del sistema?

¿Qué propiedades son relevantes para la medir si el sistema tiene calidad?

¿Se desarrolla una evaluación específica de los errores del software para poder obtener un sistema de calidad con respecto al modelo ISO 9001?

¿El sistema tiene buenas verificación y respuestas rápidas de pruebas unitarias y de integración con respecto a mejores prácticas?

Las preguntas desarrolladas las expone el desarrollador si cumple esos estándares, si esto se da se puede tener un software de calidad.

Pruebas y evaluación

Tipo de prueba:

Nivel alto: el módulo de autenticación de usuario fue evaluado por evaluadores individuales que evaluaron la funcionalidad, la capacidad de mantenimiento, la facilidad de uso y la confiabilidad del sistema. Sin embargo, esto se hizo de acuerdo con el estándar cuando se trata de casos de prueba. El módulo seleccionado fue

Nivel bajo. Son representaciones de verificación de errores de pequeños segmentos de código y son componentes fundamentales para lograr la calidad del sistema y la satisfacción del cliente. Se hizo con cambios menores para tener diferentes perspectivas de usuario. Es decir, era un módulo que fue aprobado por el personal de la Fundación e incluía la validación de campos (por ejemplo, mostrar un mensaje de error si los datos se ingresaron incorrectamente o se construyeron incorrectamente).

Aplicar técnicas y herramientas para medir la calidad

Para calificar la calidad del sistema se utilizó una herramienta que esta dentro del framework de Angular denominado spec. Este archivo se agrega a la colección de directorios cada vez que se hace referencia a un componente, servicio o página en el archivo angular.json.

Tenga en cuenta que en el archivo de especificación existe una tecnología llamada Karma que escucha los cambios usando el comando Ng Test y continuar verificando cuál es el error de la función colocada en el componente o en un servicio, este método es útil para probar fragmentos de código y comprobar que se devuelve una respuesta válida.

```
You, 2 hours ago | 2 authors (You and others)
1 import { async, inject, TestBed } from '@angular/core/testing';
2 import { LectorService } from './lector.service';
3 import {
4     HttpClientTestingModule,
5     HttpTestingController,
6 } from '@angular/common/http/testing';
7 describe('LectorService', () => {
8     let service: LectorService;
9
10    beforeEach(() => {
11        TestBed.configureTestingModule({
12            imports: [HttpClientTestingModule],
13            providers: [LectorService],
14        });
15        service = TestBed.inject(LectorService);
16    });
17
18    it("Should be created", inject([LectorService], (service: LectorService) => {
19        expect(service).toBeTruthy();
20    }));
21
22    it("should get lector", async(() => {
23        const service: LectorService = TestBed.get(LectorService);
24        service.getLectores().subscribe(
25            (response) => expect(response).not.toBeNull(),
26            (error) => fail(error)
27        );
28    }));
29 });
30 });
31
```

Ilustración 15 Archivo spec de Lector service. Fuente: Propia (2022)

Al inicio del archivo hay inyección de dependencias, en este caso se llama a la clase Lector Service, propiedades y funcionalidades de angular como: HttpClientTestingModule, async, inject, y TestBed. Se puede observar que la primera función regresa un método foreach para la respectiva llamada de las inyecciones de independencia. El segundo método hace un llamado directamente al servicio y verifica si existen atributos o funciones presentes. Por último, una funcionalidad que se caracteriza por el llamado de un método en específico añadiendo la propiedad async para tener una respuesta en el tiempo correcto, y debido a realizar una subscripción a la función al ser un observable, y regresar un expect que devuelve la respuesta de la función, que en el presente caso sería el arreglo de lectores, solicitados del API del servicio web. Consciente tener una visualización del testeado unitario a través del comando, `ng test --code-coverage --include`, se muestra la presente pantalla en el navegador.



Ilustración 16: Karma v3.1.6 Fuente: Propia (2022)

4.10 Despliegue

4.10.1 Entrega

En la fase de entrega o implementación, se considera que el software ha sido probado tantas veces como sea necesario para evitar errores o equivocaciones en la fase de producción. El software se entrega al usuario final con todos los derechos de acceso posibles, ya sea con una cuenta para acceder al sistema o de alguna forma para que el usuario pueda interactuar con el software sin problemas. La fase de producción de una aplicación web es subir el proyecto al servidor y considerar siempre las partes técnicas, como qué tipo de servidor web, cuentas o acceso para administrar e instalar paquetes o archivos. Cabe señalar que no se requiere instalación de la máquina, ya que cargar el proyecto en la máquina host, dada la dirección IP pública, significa que el software se puede usar desde cualquier máquina con conexión a Internet. Al usar el software en producción, siempre se recomienda usar la dirección IP pública del proveedor en el servicio web, ya que así es como funciona la producción y los usuarios pueden acceder al sistema.

Conclusión

Según las necesidades identificadas de la fundación Leonidas Ortega se pudo deducir un análisis de requerimientos técnicos para poder seleccionar metodologías de desarrollo, diseñar prototipos y finalmente evaluar el sistema informático según métricas de calidad de software. Las necesidades encontradas fueron de fundamental importancia para poder entender los problemas que la fundación tiene debido que no tiene un monitoreo y gestión de los lectores, es decir, se encontró que no se manejan reportes ni control de los préstamos de libros hacia los beneficiarios.

La importancia de la elección de metodología, patrones de diseño y arquitectura durante el desarrollo de software es fundamental para que el sistema sea mantenible y escalable, se optó por el desarrollo continuo de acuerdo a la metodología espiral con prototipos, la elección se dio por varias razones, una de las cuales fue la presencia del COVID 19 durante gran parte de 2020, lo que cambió otras metodologías de una reunión continua a imposible, pero la elección de una espiral fue la combinación perfecta, presentando los prototipos de cada módulo durante un período determinado que no se especificó. Según la situación del país y del mundo. Las reuniones se realizan en línea y con presentaciones según los deseos del usuario, siempre respetando la metodología propuesta, comenzando con una encuesta informativa.

Para el desarrollo del software se utilizó el framework Angular, el cual ofrece muchas ventajas para el desarrollo web, una de ellas es la reactividad, control de dependencias, buenas prácticas en la estructura de archivos y directorios. Para el desarrollo del aplicativo móvil se utilizó la tecnología Flutter la cual fue de mucha ayuda para la elaboración de interfaces graficas amigables y modernas debido a que esta herramienta usa widgets que son como piezas de legos interconectadas y se basa mucho en gestores de estado para el correcto uso de la programación funciona.

El modelo basado en componentes ayudó mucho a construir una GUI limpia y moderna, evitando la recarga innecesaria de páginas o datos, se reutilizó una gran cantidad de componentes y la GUI cumplió con el diseño, la estructura y la flexibilidad, así como con la asequibilidad y aceptación de usuario. La evaluación del software es un paso crucial en la fase de desarrollo, las métricas

de calidad correspondientes a las pruebas de modelo, unidad, integración y aceptación, hacen que el sistema sea escalable, mantenible, calidad a nivel de código y estructura, muchos parámetros fueron evaluados en la fase de prueba.

Recomendaciones

En este proyecto se aplicó un modelo en espiral combinado con prototipos, que es una metodología apta para todos los proyectos de software debido a las fases de uso del modelo, caracterizadas por análisis de riesgos, prevención de posibles errores y pruebas unitarias. Integración que ayuda a reducir los riesgos potenciales. Cuando se utiliza la metodología espiral con prototipos, se recomienda utilizar una matriz de riesgo durante todo el ciclo de vida del software, especialmente para el diseño y análisis. Esto ayuda al desarrollador a saber qué hacer cuando ocurre un conflicto, cuándo ya no puede ocurrir y qué tan probable es que ocurra un riesgo potencial. Las tecnologías como Angular y Flutter tienen muchas características que aseguran un desarrollo continuo, la curva de aprendizaje es media debido a los muchos conceptos que se cubren en el marco, reactividad, métodos asíncronos, inyección de dependencia, etc.

Para implementar las mejores prácticas de Angular y Flutter, hace que el software sea un código limpio, mantenible y escalable. Se recomienda estudiar los conceptos de manejadores de estado, métodos asíncronos y reactividad, que son estándares en la programación de la técnica propuesta. La prueba de la fase de desarrollo del software es estándar en el ciclo del sistema informático, cabe señalar que el software no debe lanzarse a producción a menos que haya sido probado o probado por algunos usuarios. Se debe considerar la aplicación de métricas de calidad, que en muchos casos corresponden a métodos cuantitativos o cualitativos, y dependiendo del modelo seleccionado, se puede asegurar el cumplimiento del sistema informático con los estándares correspondientes, mientras que la calidad debe ser siempre inherente al programador elección en la sección de prueba.

Cumplir con los parámetros durante la fase de prueba significa que el software tiene la menor cantidad de errores posible y puede corregirse antes de que el sistema entre en producción; además, los usuarios obtienen ideas para futuras consideraciones de mantenimiento. Para el desarrollo continuo del sistema de información gerencial, se recomienda utilizar un manual de usuario que integre las partes técnicas de las instalaciones y las configuraciones

necesarias para asegurar la consistencia a nivel de código. Además, se recomienda entender cada módulo del sistema, así sea el mantenimiento más pequeño, porque el sistema está relacionado con cada parte, y no entender el módulo crea problemas para el desarrollador al momento de analizar un nuevo requisito.

Bibliografía

- Dávila, A., García, C., & Córdor, S. (2016). *Proquest*. Obtenido de <https://search.proquest.com/docview/1912145515/fulltextPDF/A8CB8C624A3E4EDBPQ/1?accountid=130858>
- Alcaraz, S. H. (14 de Enero de 2014). *Escritores*. Obtenido de Sitio web de escritores: <https://www.esritores.org/recursos-para-esritores/recursos-1/colaboraciones/9874-la-importancia-de-la-lectura>
- Alejandro Manuel Rubinos Carvajal, H. A. (29 de 9 de 2011). *Redalyc*. Obtenido de <https://www.redalyc.org/articulo.oa?id=378343671005>
- Morales, P. (25 de Abril de 2009). *Tipos de variables y sus implicaciones en el diseño de una investigación*. Obtenido de studylib: <https://studylib.es/doc/6585937/tipos-de-variables-y-sus-implicaciones-en-el-dise%C3%B1o-de-una>
- Viveros, D. (s.f.). *Fundación Leonidas Ortega*. Obtenido de <https://flom.org.ec/nuestra-historia>
- Pressman, R. S. (2010). *Ingeniería del software. Un enfoque práctico*. McGrawHill.
- Sanz, L. F. (2005). *Revista Redalyc*. Obtenido de <https://www.redalyc.org/pdf/922/92210205.pdf>
- Auth0. (s.f.). *Introduction to JSON Web Tokens* . Obtenido de JWT: <https://jwt.io/introduction/>
- Martínez Villalobos, G., Camacho Sánchez, G. D., & Biancha Gutiérrez, D. A. (Abril de 2010). *Diseño De Framework Web Para El Desarrollo Dinámico De Aplicaciones*. Obtenido de <https://www.redalyc.org/pdf/849/84917316032.pdf>
- Lexis Finder. (2 de Agosto de 2010). *Ley Organica De Educacion Superior, Loes*. Obtenido de <https://www.ces.gob.ec/documentos/Normativa/LOES.pdf>
- Fundacion Leonidas Ortega. (7 de Julio de 2022). Obtenido de Flom: <https://flom.org.ec/calendario.php?id=1116>

- OEI. (14 de Junio de 2022). OEI. Obtenido de <https://oei.int/oficinas/ecuador/noticias/se-presento-los-resultados-de-la-encuesta-de-habitos-lectores-practicas-y-consumos-culturales>
- Vásquez, F. A., Paredes, S. L., & Pérez, J. F. (13 de Octubre de 2021). *Efecto de las estrategias de innovación en el éxito de las empresas*. Obtenido de RES NON VERBA REVISTA CIENTÍFICA: <https://doi.org/10.21855/resnonverba.v11i2.533>
- Montoya, S. (2009). Obtenido de Redalyc: <https://www.redalyc.org/pdf/6139/613965348012.pdf>
- Olivera, D., Caridad, L. d., Alonso, D., & Manuel, L. (2020). Obtenido de Redalyc: <https://www.redalyc.org/journal/3783/378366538003/378366538003.pdf>
- Carvajal, R., Manuel, A., León, N., & Alina, H. (2011). Obtenido de Redalyc: <https://www.redalyc.org/pdf/3783/378343671005.pdf>

Anexos

1. Levantamiento de información

Web Administrador:

- Ingreso por credenciales de acceso
- Crud de un libro
- Crud de lector
- Buscar lector.
- Buscar libros.
- Asignación de libro a lector.
- Pantalla para ver estadísticas.
- Estadística de cantidad de Libros
- Estadística de cantidad de Libros prestados
- Estadística de cantidad de Lectores
- Estadística de cantidad de Lectores leyendo
- Estadística de cantidad de Lectores que han terminado un libro
- Tabla de monitoreo de lectura (Alumno, Libro, Fecha Inicio, Tiempo Transcurrido)
- Informe de estadísticas en formato excel descargable

App móvil

- Ingreso por credenciales de acceso
- Perfil de usuario (Datos básicos)

- Edición de datos de perfil
- Sección de Libros prestados
- Función de dar por finalizado un libro
- Sección de Libros Leídos

2. Pruebas con usuario



Ilustración 17: Reunión sobre prototipos



Ilustración 18: Reunión sobre requerimientos



Ilustración 19: Reunión sobre validaciones de la app



Ilustración 20: Reunión sobre funcionalidad de app

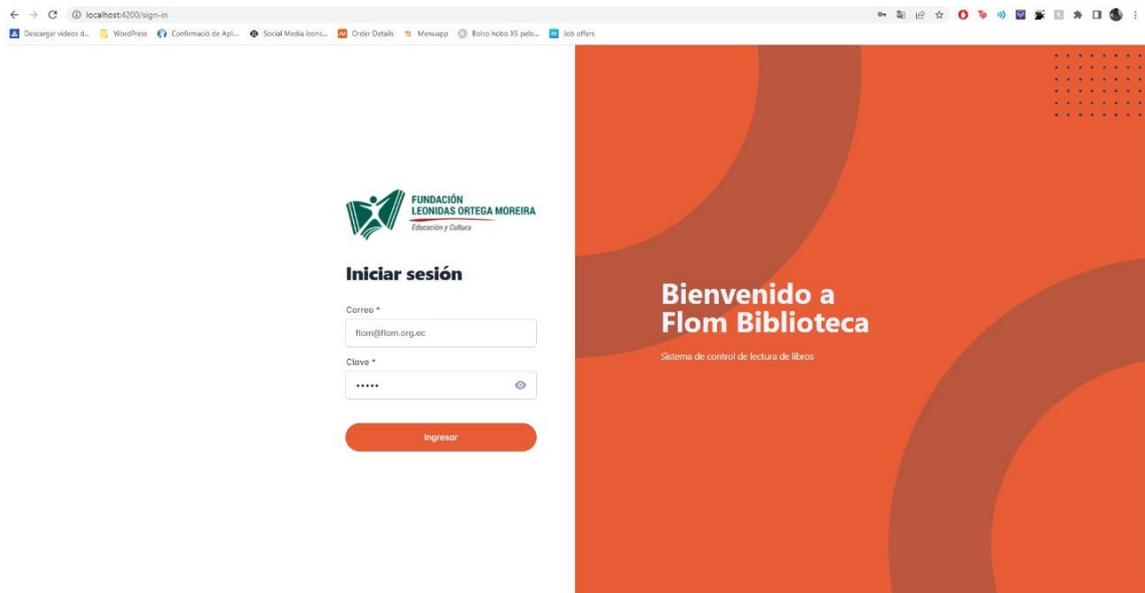


Ilustración 21 Login del administrador. Fuente: Propia (2022)

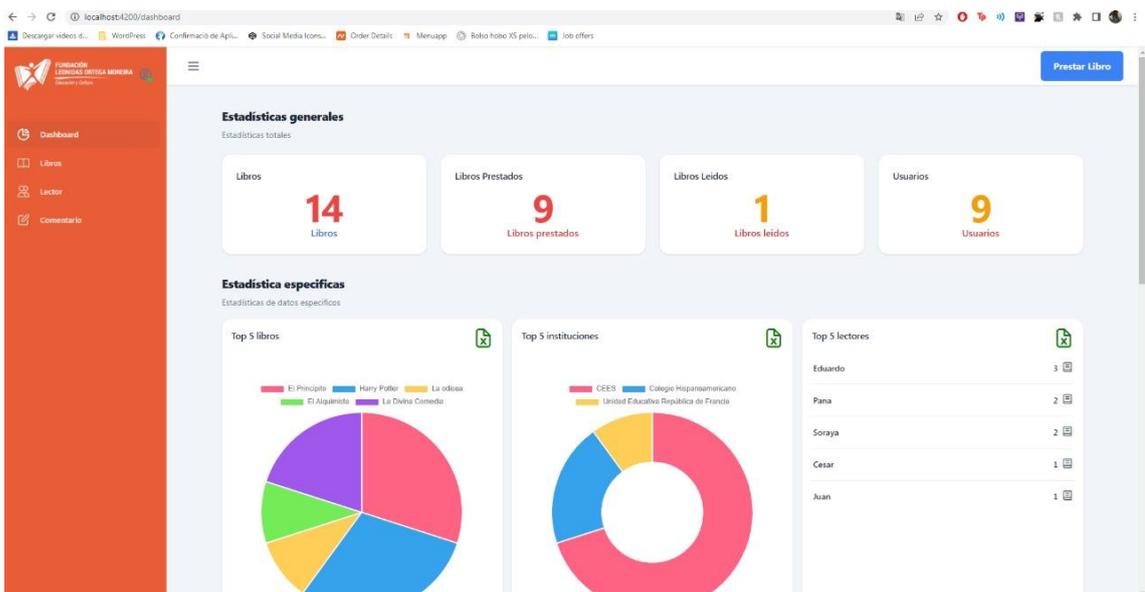


Ilustración 22 Modulo de Estadísticas Generales y específicas. Fuente: Propia (2022)

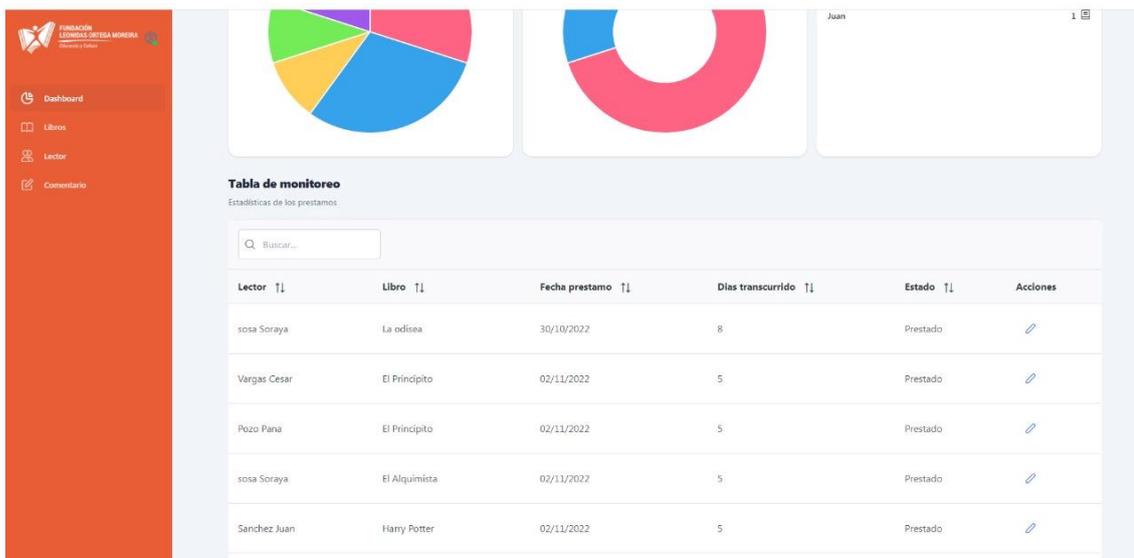


Ilustración 23 Modulo Estadísticas tabla de monitoreo. Fuente: Propia (2022)

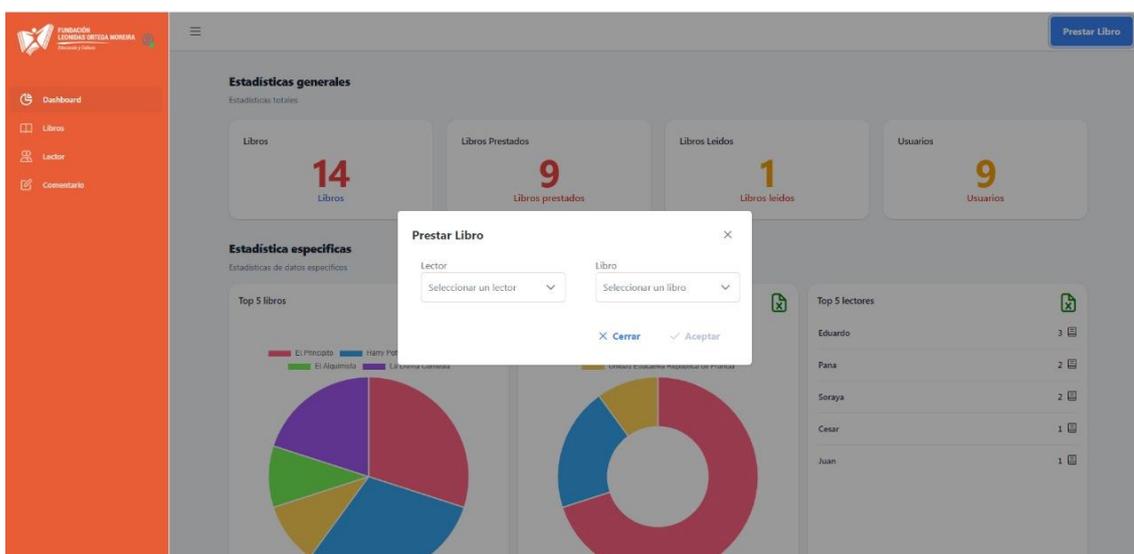


Ilustración 24 Funcionalidad préstamo libro. Fuente: Propia (2022)

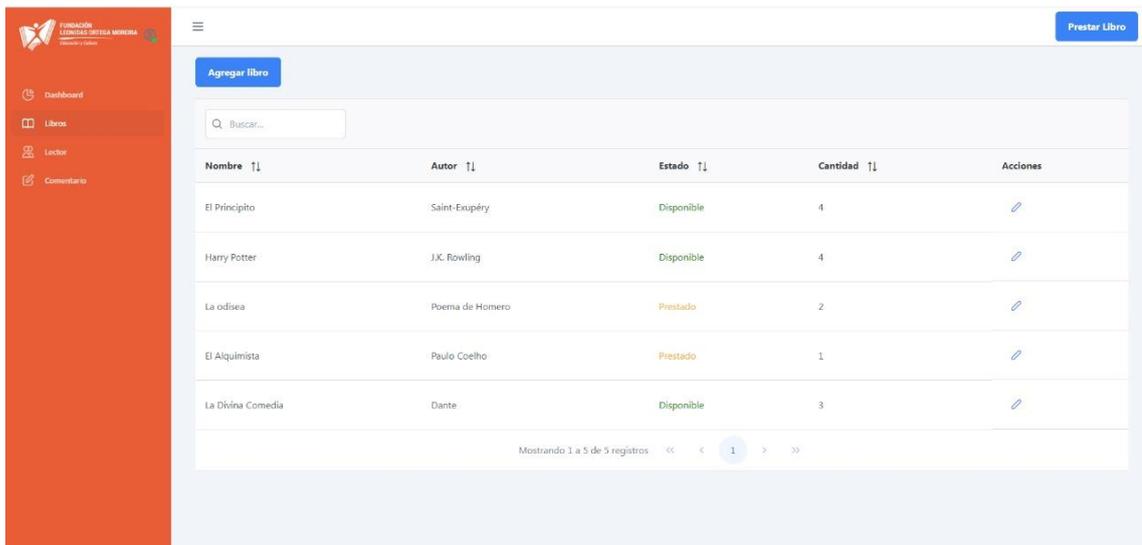


Ilustración 25 Modulo Libro. Fuente: Propia (2022)

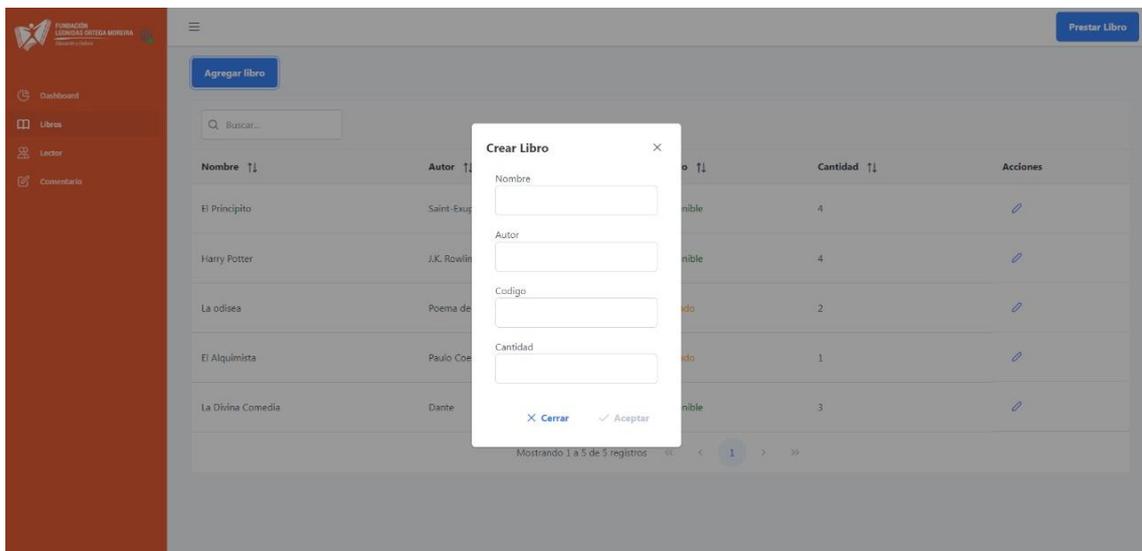


Ilustración 26 Funcionalidad crear libro. Fuente: Propia (2022)

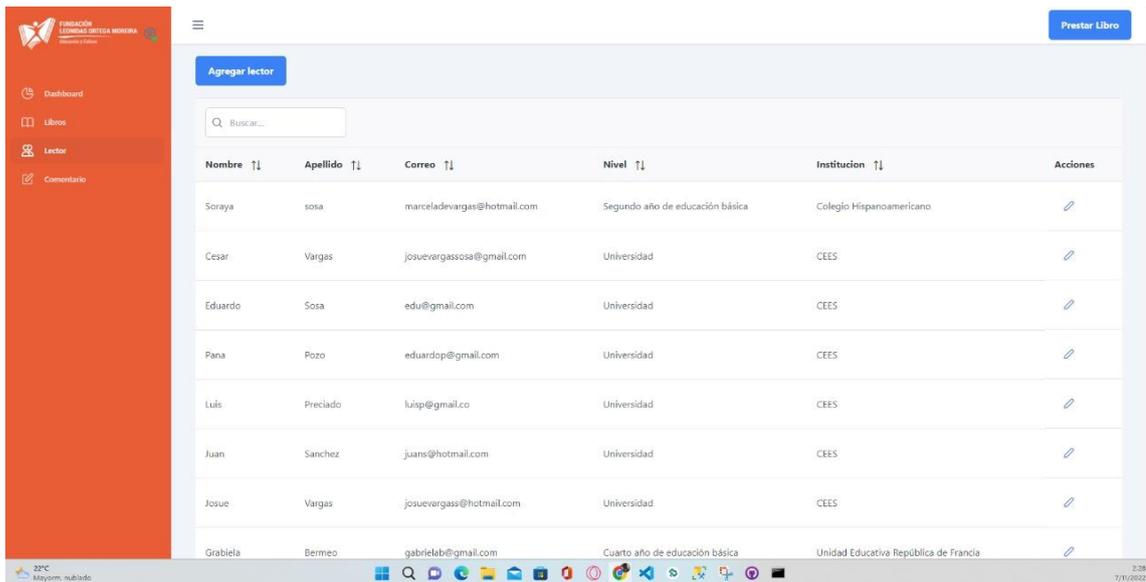


Ilustración 27 Modulo Lector. Fuente: Propia (2022)

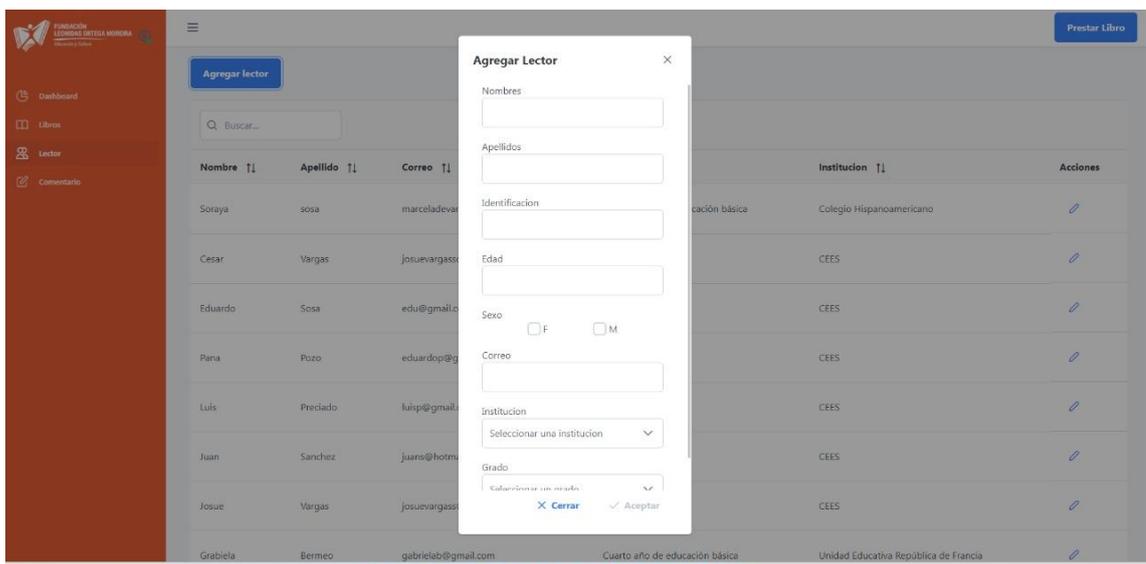


Ilustración 28 Funcionalidad agregar lector. Fuente: Propia (2022)

FUNDACIÓN ESCUELA CRISTICA MOREIRA
 Educación y Cultura

Dashboard
 Libros
 Lector
 Comentario

Prestar Libro

Lector ↑↓	Libro ↑↓	Fecha envio ↑↓	Descripción
Eduardo	El Principito	Nov 6, 2022	Esto es una prueba de un comentario del préstamo con el id 2009 Esto es una prueba de un comentario del préstamo con el id 2009 con el id 2009 con comentario del préstamo con el id 2009 Esto es una prueba de un comentario del préstamo con el id 2009 Esto es una prueba de un comentario
Eduardo	Harry Potter	Nov 6, 2022	Este es el segundo comentario de prueba

Mostrando 1 a 2 de 2 registros << < 1 > >>

Ilustración 29 Modulo comentarios Fuente: Propia (2022)

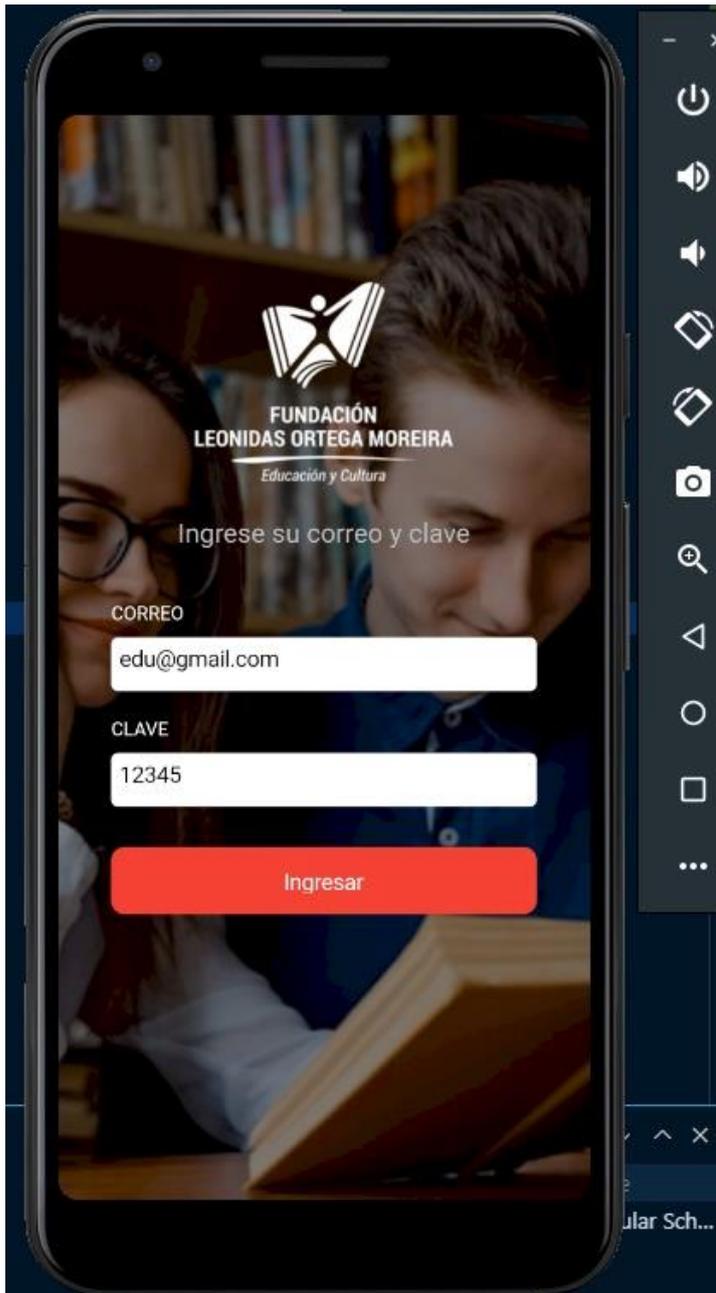


Ilustración 30 Login App Móvil Fuente: Propia (2022)

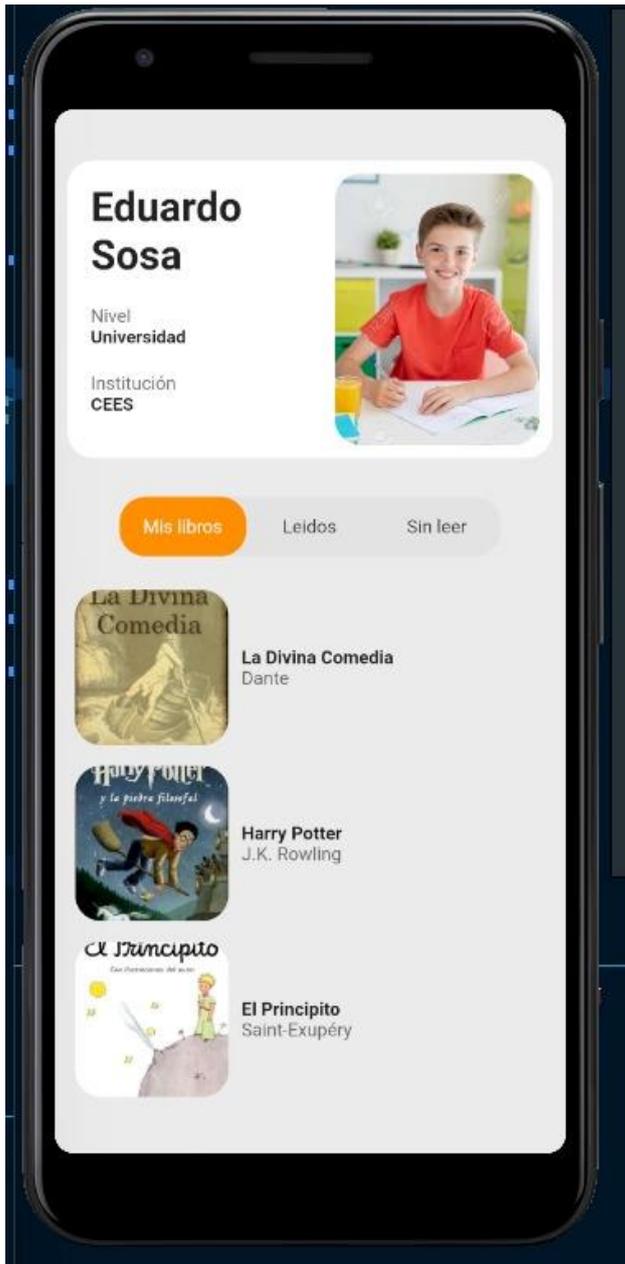


Ilustración 31 Perfil Lector. Fuente: Propia (2022)

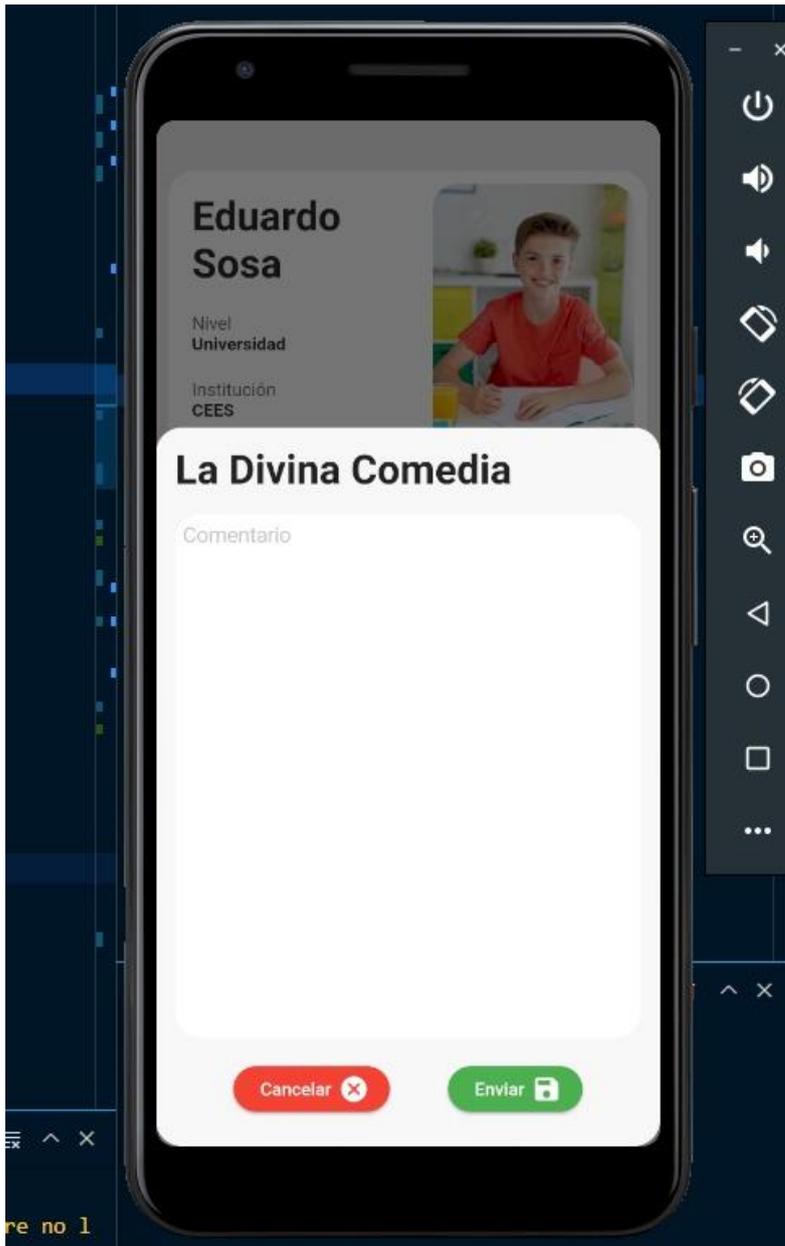


Ilustración 32 Funcionalidad comentar libro Fuente: Propia (2022)