

UNIVERSIDAD ESTATAL A DISTANCIA
VICERRECTORIA ACADEMICA
ESCUELA DE CIENCIAS EXACTAS Y NATURALES
CARRERA INGENIERÍA INFORMATICA

PROYECTO No 1

MODALIDAD ESCOGIDA: PROYECTO

Grupo: 02

PROYECTO #1 PARA EL CURSO
DE ANALISIS Y DISEÑO DE SISTEMAS

PABLO ANDRÉ VALENCIANO BLANCO

1-1572-0043

Profesor: Jorge Alvarado Zamora

CENTRO UNIVERSITARIO DE HERERIA

PAC: 2025-2

Fecha Entrega: 22 de Junio 2025

CIUDAD: HEREDIA

Contenido

INTRODUCCIÓN	1
Parte 1. REQUERIMIENTOS EN EL DESARROLLO DE SOFTWARE	2
Sección A. Técnicas de levantamiento de requisitos: Entrevista y Prototipado.....	2
Sección B. Preguntas cerradas en busca de funcionalidad móvil.	3
Sección C. Importancia del levantamiento de requerimientos.....	4
Parte 2. PRÁCTICAS DE SOSTENIBILIDAD Y EFICENCIA ENERGETICA.....	5
Sección A. Cuadro de posibles prácticas.	5
Sección B. Importancia.	5
Parte 3. CASOS DE USO	7
Sección A. Registrar usuario/lector afiliado.....	7
Sección B. Registrar préstamo.	9
Sección C. Registrar pago.....	11
Sección D. Enviar notificación de disponibilidad de libro.....	12
Sección E. Enviar recordatorio de devolución de libro	13
Parte 4. DIAGRAMAS DE CASOS DE USO	14
Parte 5. MODELO DE NEGOCIO -> Modelo DOMINIO	15
Sección A. Modelo Dominio, así es como funciona.	15
Sección B. Aplicación sobre el caso de estudio	15
CONCLUSIÓN.....	17
REFERENCIAS	18

Tabla de Figuras

Figura 1.....	14
---------------	----

Tabla de Tablas

Tabla 1.....	3
Tabla 2.....	5
Tabla 3.....	16

INTRODUCCIÓN

El siguiente documento presenta la realización del primer proyecto del curso Análisis y Diseño de Sistemas realizado por el estudiante Pablo André Valenciano Blanco en lo que es el segundo cuatrimestre del año 2025, los proyectos siguen los lineamientos del documento descrito por trabajos de informática y utilizando las referencias APA de séptima edición.

Este primer proyecto consta de cinco partes, donde cada uno de estas poseerá una cantidad variable de secciones, donde lo mínimo van a ser dos y lo máximo una cantidad de cinco. Cada una de estas partes relata un poco sobre lo visto durante los capítulos de la unidad didáctica Ingeniería de Software (2015) desde el capítulo siete hasta el doce, donde en forma general, alineado al objetivo de este proyecto (2025) va a tener la tarea de: “El estudiante explorará los conceptos y técnicas estudiadas en el curso sobre el levantamiento, análisis, y prueba de los requerimientos, y el lenguaje UML.” Y su aplicación para el caso de estudio en el desarrollo de la solución de software Red de Bibliotecas Interconectadas (A partir de ahora, se le nombrará como RedBiln)

A modo introductorio, la primera parte consta de explicar aquellas técnicas que sirven para levantar requisitos, en especial las de Encuesta y Prototipado, conociendo su importancia y su posible uso en el caso RedBiln. La segunda parte del proyecto se discute las posibilidades de sostenibilidad y responsabilidad ambiental del caso RedBiln y la importancia que tiene hacer este tipo de análisis. La siguiente parte es para ejemplificar el uso de los casos de uso, en cinco funciones del caso de estudio que se desean implementar para el proyecto, siguiendo las instrucciones de la unidad didáctica. En la cuarta parte se hace el diseño de los diagramas de casos para cada uno de los actores implicados en el uso de la solución de software. Por último, se habla del Modelo de negocio, específicamente el de Dominio y como este se aplica sobre el caso de estudio. Para dar por concluido el documento se brindan 5 conclusiones referentes al tema, donde cada una de estas involucra lo aprendido y dominio de cada una de las partes de la cual esta se compone.

Vamos a dar inicio al marco teórico...

Parte 1. REQUERIMIENTOS EN EL DESARROLLO DE SOFTWARE

Sección A. Técnicas de levantamiento de requisitos: Entrevista y Prototipado.

La definición de Prototipado que ofrece Schork y Kirchner (2018) es la misma que “describe el prototipado como una necesidad general, pero no ofrece enfoques metodológicos para el desarrollo de los propios prototipos con el objetivo de obtener el máximo conocimiento. Por tanto, el prototipado se realiza principalmente de forma intuitiva e iterativa, lo que a menudo conduce a un proceso ineficiente.”

El prototipado según menciona Gomaa y Scott (1981), para el levantamiento de requisitos se tiene que: “Aunque el prototipado suele considerarse costoso, corregir ambigüedades y malentendidos en la etapa de especificación resulta considerablemente más económico que corregir un sistema una vez que ha entrado en producción.”

El desarrollador de software que utilice esta técnica, cual cita Schork y Kirchner (2018), el desarrollador debe tener en cuenta que el proceso de fabricación del prototipo puede diferir del proceso del producto final, lo cual también conlleva requisitos diferentes para el prototipo.

Belani, Pripuzić y Kobaš (2005) analizaron el uso de encuestas como “un objetivo fundamental en la ingeniería de requisitos es descubrir las verdaderas necesidades de los interesados. Realizar una buena encuesta permite obtener la información adecuada de estos participantes.”

Las encuestas son esenciales para la construcción y definición del problema tal como enmarca Ghai y Kaur (2012). Donde se destaca que en muchas ocasiones provee la suficiente información donde se puede llegar a prever el alcance temporal y espacial y la importancia de dar una posible solución a la problemática asociada que el equipo de desarrollo de afrontar, como dar un análisis preliminar de las prioridades de desarrollo, como las funcionalidades que se podrían aplazar al tiempo.

Sección B. Preguntas cerradas en busca de funcionalidad móvil.

Pregunta 1. ¿Tal como se maneja las bibliotecas a nivel Nacional a día de hoy, es una labor suficiente? R/ Si o no.	Pregunta 6. ¿Estaría usted interesado en tener la posibilidad de que la devolución de un libro pueda hacerlo no solo en la biblioteca que retiro sino también en alguna otra que le quede más cómoda? R/ Si o no.
Pregunta 2. ¿Al pedir prestado Libros durante este año haz hecho uso de cuantas Bibliotecas de la GAM? R/ Si o no	Pregunta 7. Este servicio podría llegar a solicitar un monto de afiliación anual, entre estas opciones ¿Cual consideras un monto justo a pagar para hacer uso de los beneficios que RedBiln? R/ Intervalos de pago.
Pregunta 3. ¿En estos 12 meses, cuantos libros has solicitado de préstamo en una de las Bibliotecas de la GAM? R/ Numérica u opciones con Intervalo	Pregunta 8. La aplicación móvil desea proveer un mapa que le indique al usuario la cercanía de una Biblioteca Interconectada Del 1 al 5, donde 1 es lo más bajo y 5 más alto. ¿Es esto algo útil para usted como cliente? Considere la existencia de Waze y Maps.
Pregunta 4. ¿Está tu biblioteca de preferencia dentro de las 30 bibliotecas a interconectar? R/ Si o no.	Pregunta 9. ¿Estarías dispuesto(a) a cambiar los tramites físicos (efectivo/tarjeta) por tramites completamente digitales? R/So-No
Pregunta 5. En una escala de 1 a 5, donde 1 es la calificación más baja y 5 la más alta, ¿cómo calificaría usted su disposición a usar aplicaciones móviles para tramitar servicios? R/ Escala de 1 a 5	Pregunta 10. ¿De entre cuantos días estarías dispuesto(a) a esperar, desde que solicita un libro de otra biblioteca hasta que llegue a la de tu preferencia? R/ Numérica o Intervalos

Tabla 1. Preguntas cerradas para ejemplo de encuesta para caso de estudio

Sección C. Importancia del levantamiento de requerimientos.

En la presente sección se habla con palabras propias del autor del documento según lo que se ha logrado aprender con respecto a la utilidad del levantamiento de requisitos, donde la unidad de Panteleo (2015) ha sido pieza esencial para la adquisición del conocimiento, la cual no se citara para mantener la idea del proyecto, esta pequeña introducción no forma parte de las líneas solicitadas.

En palabras propias, los requisitos son tan importantes como lo son las ventas finales, a nivel de costos. Esto ya que la forma como se definen, serán la piedra angular de la solución de software a desarrollar, brindando tanto la delimitación del problema a resolver, el alcance que va a tener el proyecto, que los involucrados al proyecto “defiendan con espada y escudo” los ideales de lo que representa la solución y finalmente dan la confianza a los inversionistas que introducen su capital, serán los requisitos tan duros o frágiles, según como se haya “alimentado” con capital y personal humano para definir cada uno de estos.

Muchas empresas lastimosamente, por una supuesta emergencia del mercado, suelen reducir en gran medida los recursos, especialmente tiempo, al ser una etapa que según los que no saben cómo un gasto sin producción, tal como un pasivo similar al alquiler de un local. Sin considerar que al hacer esto de manera correcta, pueden llegar a generar 10 veces más al valor de la solución y ahorrarse muchos gastos de una correcta preparación.

En fin, a opinión del autor, no significa que se deba pasar 2 o 3 años levantando requisitos, ya que a fin de cuentas el negocio se hace para invertir en recurso humano y tecnológico para generar insumos, sino que desde la misma etapa de planificación con un agente experto se definan los limites, condicionales y dar un tiempo prudente para analizar las conclusiones que las herramientas como la encuesta y el prototipado ofrecen, a veces pasa que gracias a esto, se ha ahorrado en invertir millones en negocios sin un problema correcto sin definir.

Parte 2. PRÁCTICAS DE SOSTENIBILIDAD Y EFICIENCIA ENERGETICA

Sección A. Cuadro de posibles prácticas.

Prácticas de sostenibilidad y eficiencia energética	Objetivo que se busca con esta práctica
1. Refresco de Libros a la base de datos del servidor debe ser hecha cada 3 horas, en vez de tiempo real.	Las consultas con cantidades de miles de filas duran entre 20 a 40 minutos, al realizar un refresco periódico se normaliza a nivel de usuarios y administradores el tiempo que la información se muestra en pantalla.
2. Cada biblioteca, en un minicomputador tendrá los datos propios de información (Los libros que estén solamente en su biblioteca)	En caso de un hackeo o pérdida información, se posea un respaldo y se pueda verificar entre servidor y base de datos interna, para hacer correcciones.
3. Rutas de transporte de Libro, solo ante pedido.	Evitar pagar y gastar combustible, para rotar libros que no se han solicitado, solo por el deseo de promoción. Definir las rutas de transporte, que consuman menos distancia y posibles rutas alternas.
4. Envío de libros en paquetes. No individual.	Solo se trasladarán cuando se alcance una meta de libros a trasladar. No usar recursos en cantidades minoristas y dar un uso de al menos 30% de ahorro.
5. Apagado de funcionalidades no críticas fuera de horarios de oficina	Al momento de que las oficinas cierren, se espera que aquellas funciones que piden el ingreso de administradores no se utilicen, por lo que las maquinas que estas acceden se apaguen.

Tabla 2. Prácticas de sostenibilidad y eficiencia energética para el caso de estudio.

Sección B. Importancia.

La eficiencia energética en el software está estrechamente ligada a la reducción de costos operativos. Un sistema que consume menos energía genera menos gastos eléctricos, especialmente relevante en infraestructuras a gran escala como los centros de datos. Además, el software eficiente puede reducir la necesidad de actualizaciones de hardware frecuentes, lo cual también disminuye los residuos electrónicos. Según Koçak et al. (2020), “la eficiencia en el uso de recursos permite una reducción sustancial en el TCO (Total Cost of Ownership) de los sistemas informáticos”.

Acorde a Bozelli (2021) la responsabilidad social y la reputación corporativa debe ser importante para las empresas al momento de “adaptarse a las prácticas sostenibles y

promueven la eficiencia energética en sus sistemas informáticos mejoran su reputación y fortalecen su compromiso con la responsabilidad social corporativa. Esto puede traducirse en ventajas competitivas, mayor fidelidad de los clientes, atracción de talento y alineación con los valores de las nuevas generaciones de usuarios más conscientes del medio ambiente.”

Parte 3. CASOS DE USO

Sección A. Registrar usuario/lector afiliado.

Caso de Uso: Registrar usuario/lector afiliado.

Descripción del caso de uso: Permite registrar al usuario y afiliar lectores a la Red.

Actores: Usuario para crear cuenta, un empleado puede crearlo como ayuda.

Precondiciones: Sin precondiciones

Activación: Se activa al ingresar a la app móvil y no tener cuenta y el empleado podrá crear usuarios para ayudar a clientes personalmente.

Poscondiciones:

Campo	Tipo	Obligatorio	Valor defecto	Solo Lectura	Tipo de componente
Identificación	Numérico	Si	-	No	Caja de Texto
Nombre Completo	Texto	Si	-	No	Caja de Texto
Afiliación	Booleano	No	No	No	Caja de chequeo
Preferencia de Pago	Texto	No	-	No	Combo

Flujo Principal:

1. La persona ingresa con el correo.
2. Si el correo no tiene cuenta, le solicita el formulario con los datos.
3. Correo de confirmación.
4. Termina de crear la cuenta

Flujo Alternativo:

1. La persona ya tiene cuenta, lo mando a la página de inicio.
2. Al dar check en Afiliación, le pedirá preferencia de pago, donde podrá incluir Simpe o pago en Línea

Flujo de Validación:

1. Los campos obligatorios deben llenarse, si no deberá darse.
2. Se debe dar una identificación válida y que no se haya repetido en otra cuenta.

Sección B. Registrar préstamo.

Caso de Uso: Registrar préstamo.

Descripción del caso de uso: Permite registrar el préstamo de uno o varios libros por parte de un usuario afiliado en cualquier biblioteca de la RedBiln, ya sea de forma presencial o digital.

Actores: Usuario Afiliado, Bibliotecario, Sistema RedBiln

Precondiciones: El usuario debe estar afiliado y tener la membresía activa, el libro debe estar disponible.

Activación: El usuario solicita el préstamo de un libro desde la plataforma web/móvil o en ventanilla.

Poscondiciones: El préstamo queda registrado en el sistema, se actualiza el estado del libro como prestado, se asocia una fecha límite de devolución.

Flujo Principal:

1. El usuario accede al catálogo o solicita el libro.
2. El sistema verifica la disponibilidad.
3. El bibliotecario (o sistema) registra el préstamo.
4. El sistema actualiza el inventario.
5. Se genera un comprobante y se notifica al usuario.

Flujos Alternativos:

1. Si el libro está en otra biblioteca, se programa un préstamo interbibliotecario.
2. Si hay una reserva previa, se cancela el préstamo.

Flujo de Validación:

1. Verificar afiliación y estado del usuario.
2. Validar disponibilidad del libro.
3. Validar que el usuario no tenga préstamos vencidos.

Sección C. Registrar pago.

Caso de Uso: Registrar pago.

Descripción del caso de uso: Permite registrar pagos por afiliación, préstamo, reserva o multas desde la aplicación móvil, página web o ventanilla.

Actores: Usuario afiliado, Sistema de pago, Bibliotecario (en caso presencial)

Precondiciones: El usuario debe estar autenticado en el sistema, debe haber un monto pendiente de pago.

Activación: El usuario realiza una acción que requiere un pago o accede a su historial para saldar deudas.

Poscondiciones: El pago queda registrado, se actualiza el estado de la deuda a “pagado”. se genera un comprobante y se notifica al usuario.

Flujo Principal:

1. El sistema muestra el monto a pagar.
2. El usuario selecciona el método de pago.
3. El sistema procesa el pago.
4. Se confirma la transacción y se notifica al usuario.

Flujos Alternativos:

1. El pago se realiza de forma presencial y lo registra un bibliotecario.
2. El usuario rechaza la transacción o no completa el proceso.

Flujos de Validación:

1. Verificar existencia de deuda o monto a pagar.
2. Validar los datos de la tarjeta u otro medio de pago.
3. Validar conexión segura con el proveedor de pagos.

Sección D. Enviar notificación de disponibilidad de libro.

Caso de Uso: Enviar notificación de disponibilidad de libro.

Descripción del caso de uso: Permite notificar automáticamente a los usuarios que reservaron un libro cuando este se encuentra disponible en alguna biblioteca de la red.

Actores: Sistema RedBiln, Usuario afiliado

Precondiciones: El libro debe estar registrado en el sistema como “disponible”, el usuario debe tener una reserva activa para ese libro.

Activación: El sistema detecta que un libro previamente reservado está disponible.

Poscondiciones: Se envía la notificación al usuario, se marca la reserva como “disponible para retiro”.

Flujo Principal:

1. El sistema actualiza el estado del libro a “disponible”.
2. Se consulta la lista de reservas pendientes.
3. Se envía la notificación por correo y/o app.
4. Se actualiza el estado de la reserva.

Flujos Alternativos:

1. Si el usuario no responde en X días, se notifica al siguiente en la lista de espera.

Flujos de Validación:

1. Validar que el usuario tenga notificaciones activadas.
2. Validar que el libro corresponda a la reserva.

Sección E. Enviar recordatorio de devolución de libro

Caso de Uso: Enviar recordatorio de devolución de libro

Descripción del caso de uso: Permite enviar al usuario afiliado un recordatorio automático sobre la fecha de devolución de un libro prestado.

Actores: Sistema RedBiln, Usuario afiliado.

Precondiciones: El usuario tiene un préstamo activo, la fecha de devolución se aproxima.

Activación: El sistema detecta que la fecha de devolución está dentro del umbral predefinido.

Poscondiciones: Se envía el recordatorio al usuario, el usuario puede renovar o prepararse para devolver el libro.

Flujo Principal:

1. El sistema revisa los préstamos activos.
2. Identifica las fechas próximas a devolución.
3. Envía recordatorio por correo y/o notificación móvil.

Flujos Alternativos:

1. Si el usuario ya devolvió el libro, no se envía notificación.
2. Si el usuario renovó el préstamo, se reprograma el recordatorio.

Flujos de Validación:

1. Validar que el préstamo sigue activo.
2. Validar que el usuario no haya desactivado notificaciones.

Parte 4. DIAGRAMAS DE CASOS DE USO

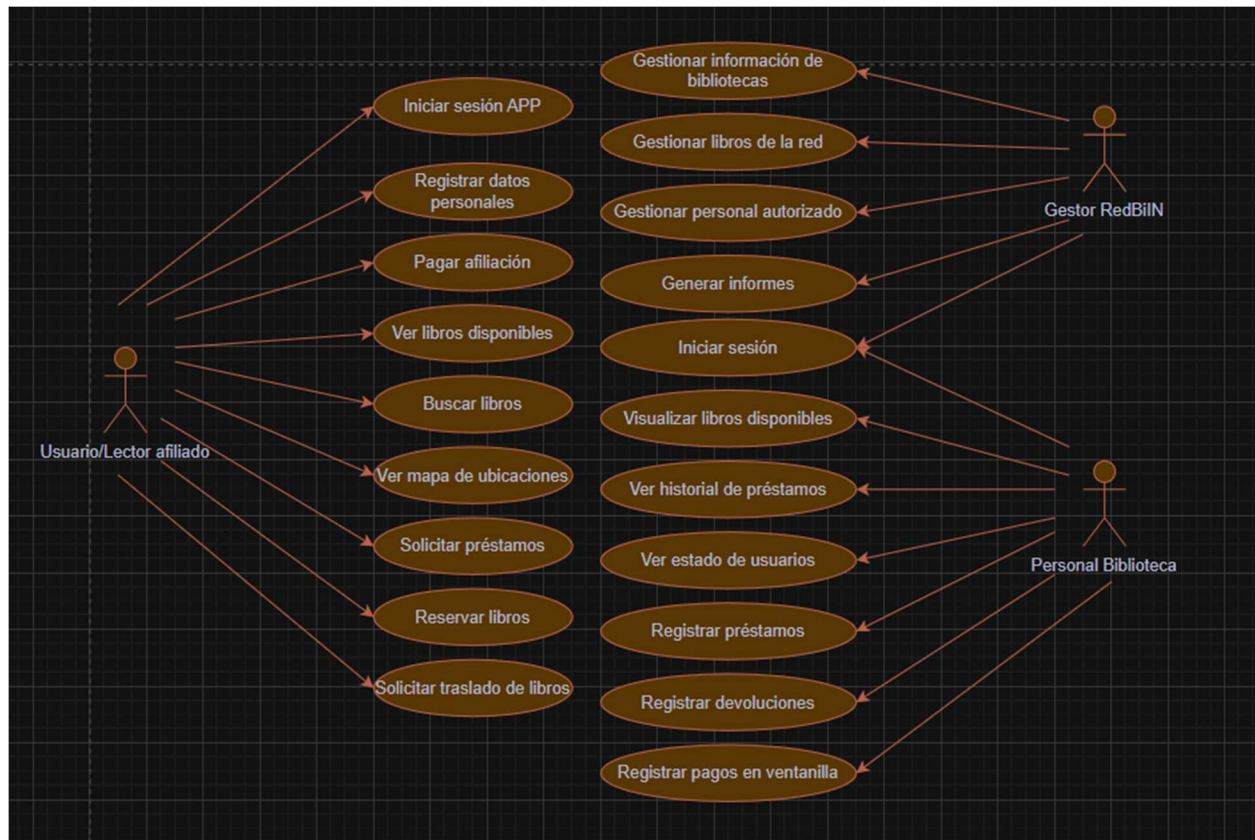


Figura 1. Diagrama de casos de Uso. Herramienta: <https://app.diagrams.net/#>

Parte 5. MODELO DE NEGOCIO -> Modelo DOMINIO

Sección A. Modelo Dominio, así es como funciona.

El modelo de dominio es una representación conceptual del conocimiento y actividades de un dominio específico. Se utiliza ampliamente en el desarrollo de software orientado a objetos y en la ingeniería del conocimiento para estructurar y entender los elementos clave de un entorno de aplicación. El modelo actúa como un puente entre el mundo real y los sistemas informáticos, permitiendo representar entidades, relaciones, reglas de negocio y comportamientos fundamentales del dominio en cuestión.

Según Campos (2004), “la modelización de dominios de conocimiento busca identificar los principios fundamentales que permiten representar el conocimiento de manera sistemática y coherente, promoviendo la reutilización, la comunicación entre expertos y desarrolladores, y la formalización del conocimiento tácito.” Campos destaca que un buen modelo de dominio no solo describe entidades, sino que capta las intenciones, restricciones y estructuras lógicas subyacentes que rigen un dominio de conocimiento.

El uso del Modelo de Dominio también es clave en áreas como la ingeniería del conocimiento, la bibliotecología y la inteligencia artificial, donde la representación conceptual del conocimiento es crucial para construir sistemas expertos, ontologías o bases de datos semánticas.

Sección B. Aplicación sobre el caso de estudio

Entidad	Atributos	Relaciones
1. Usuario/Lector	ID lector Nombre teléfono Dirección Correo electrónico Estado de afiliación Historial de préstamos	Realiza Préstamos Realiza Reservas Realiza Pagos Solicita Traslados Se afilia a una o más Bibliotecas
2. Biblioteca	ID biblioteca Nombre Descripción Ubicación Horarios Servicios Lista de Libros Cantidad de libros	Ofrece Préstamos Almacena Libros Procesa Pagos Con otras Bibliotecas mediante Préstamos interbibliotecarios

3. Personal Biblioteca	ID personal Nombre Cargo Biblioteca asignada Credenciales de acceso	Administra la Biblioteca Gestiona Libros, Préstamos, Reservas y Pagos
4. Préstamo	ID préstamo Fecha inicio Fecha fin Estado (activo/inactivo) Renovación Usuario solicitante Biblioteca origen y destino	Involucra al Usuario con los libros y Bibliotecas Puede generar Pagos o Multas
5. Libro	ID libro Título Autor Género Editorial ISBN Cantidad disponible Ubicación física Estado (disponible/prestado/reservado)	Pertenece a una Biblioteca Es objeto de Préstamos y Reservas por parte de Usuarios
6. Pago	ID pago Monto Tipo de pago (afiliación, préstamo, multa) Método de pago Fecha Estado del pago	Está asociado a un Usuario Prestamos Reservas Afiliación

Tabla 3. Modelo de Dominio aplicado al caso de estudio.

CONCLUSIÓN

Dara inicio las conclusiones para cada uno de los temas que han sido presentados a lo largo de este primer proyecto, donde el primer contacto que se logra discutir fue el uso de las herramientas que ayudan a dar el levantamiento de requisitos para una solución de software, los cuales estos deben valorizarse por su importancia al momento de delimitar el problema, definir prioridades y reconocer las limitaciones técnicas y propias de la empresa que desea usar de su tiempo para dar una solución efectiva y real.

La segunda parte se visualizan aquellas prácticas de sostenibilidad y eficiencia energética, como las empresas de los últimos años deben velar por un correcto uso de los recursos y no solo pensar en optimización del software o exigencias de los clientes, los cuales en algunos casos obvian por desconocimiento del mismo, se intentó dar 5 ejemplos para caso de estudio, donde en resumen lo que se busca es que se mantenga por prioridad la integridad de la información y que no se genere una marea constante de datos entre los servidores.

La siguiente conclusión, nos pone a piel del equipo de desarrollo generando casos de uso, para diferentes funcionalidades no todas, donde se intentó buscar los prerrequisitos, como se llegan a estos y sus posibles vías de interacción de manera tal, que no queda ninguna opción sin observar. Este no fue un trabajo sencillo ya que debe hacer todo lo posible de no poseer ninguna ambigüedad a pesar de en cierto modo ser lógico, para otros puede que sea necesario especificar.

Conectado a ello, los actores deben conocer como estos interactúan con las diferentes funciones que el software desea intervenir y que opciones ofrece dentro de la solución de diseño, para ello es necesaria la creación de lo que se conoce diagrama de casos de uso. Estos diagramas se construyen en una etapa temprano del proyecto como cajas negras que permiten ver que o quien tiene acceso a realizar acciones sobre ciertos y como lo hace.

Por último, el modelo de Dominio, no es más que un modelo de negocio que reinterpreta la comunicación de los desarrolladores, trayendo conceptos de la computación, que para gente fuera del tema, son abstractos a un ambiente más amigable con todos, permitiendo

facilidad de separar el modulado de cada funcionalidad, sin necesidad de indicar la problemática de algún otro, agilizando en gran medida el negocio y reduciendo conflictos al momento de pasar una idea al software que se esté usando para el desarrollo del proyecto.

REFERENCIAS

- Belani, H., Pripuzić, K., & Kobaš, K. (2005, June 15). *Implementing web-surveys for software requirements elicitation*. En **Proceedings of the 8th International Conference on Telecommunications (ConTEL 2005)**. IEEE. <https://doi.org/10.1109/CONTEL.2005.185931>
- Bozzelli, P. (2021). *Why Green Software Engineering Matters to Your Business*. Thoughtworks. <https://www.thoughtworks.com/insights/blog/green-software-engineering>
- Campos, M. M. (2004). *Modelização de domínios de conhecimento: uma investigação de princípios fundamentais*. *Ciência da Informação*, 33(1), 66–78. <https://doi.org/10.1590/S0100-19652004000100003>
- Gomaa, H., & Scott, D. (1981, March 9). *Prototyping as a tool in the specification of user requirements*. En **International Conference on Software Engineering**. Corpus ID: 12507561.
- Ghai, S., & Kaur, J. (2012, November). *Analysis of user requirements gathering practices in agile and non-agile software development team*. **International Journal of Computer Applications**, 58(8), 13–18.
- Koçak, S. A., Yılmaz, Y., & Aydın, M. N. (2020). *A Framework for Energy-Efficient Software Development*. *Journal of Systems and Software*, 167, 110604. <https://doi.org/10.1016/j.jss.2020.110604>
- Pantelon, G. & Rinaudo, L. (2015). *Ingeniería de Software*. 1ed. Buenos Aires, Argentina.
- Schork, S., & Kirchner, E. (2018). *Defining requirements in prototyping: The holistic prototype and process development* [Artículo]. Corpus ID: 88486272.
- Universidad Estatal a Distancia. (2025). *Caso de estudio para proyectos: Red de Bibliotecas Interconectadas (RedBiln)* [Manuscrito no publicado]. Cátedra de Ingeniería de Software, Escuela de Ciencias Exactas y Naturales, Carrera Ingeniería Informática.