INFORMÁTICA Y TELECOMUNICACIONES

UNIVERSIDAD TECNOLÓGICA DE CHILE - INACAP

SEDE SANTIAGO CENTRO

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN INFORMÁTICA

"Arquitectura de Servicios para la Gestión de Garantías mediante Notificaciones Calendarizadas "

Asignatura: Seminario de Grado – TIHI12

Sección: TIHI12

Académico guía: Ernesto Vivanco Tapia

Integrantes del equipo: Renzo Alfonso Gomez León

25 de mayo de 2025

Contenido

I. Introducción

El presente informe da continuidad al proyecto titulado “Recooba: Arquitectura de Servicios para la Gestión de Garantías mediante Notificaciones Calendarizadas”, una propuesta tecnológica desarrollada en el marco de la asignatura Seminario de Grado – TIHI12. Este proyecto surge de la necesidad de resolver un problema cotidiano y ampliamente extendido en el uso de productos tecnológicos: la pérdida, deterioro o desorganización de comprobantes físicos de compra, indispensables para ejercer derechos legales de garantía.

A través del diseño de una arquitectura basada en servicios API REST, el proyecto propone una solución digital modular, segura y escalable que permite a los usuarios registrar, almacenar y gestionar electrónicamente sus garantías, recibiendo alertas oportunas antes de su vencimiento. Esta iniciativa se enmarca en una estrategia de transformación digital orientada a la trazabilidad, automatización y respaldo documental en la nube.

En este segundo informe se detallan los aspectos de factibilidad técnica, económica y legal, así como el diseño de la solución, el desarrollo del producto, los mecanismos de aseguramiento de calidad, y la planificación de su implementación. El objetivo es documentar integralmente el proceso de construcción del sistema, desde su concepción hasta su validación en un entorno controlado.

II. Factibilidad Propuesta de Solución

1. Factibilidad técnica

La solución propuesta es técnicamente viable, ya que se basa en tecnologías maduras y de amplio uso como API REST en PHP, bases de datos MariaDB, almacenamiento en la nube y servicios de hosting web. Estas herramientas ofrecen estabilidad, escalabilidad y compatibilidad con múltiples plataformas.

El desarrollo contempla una arquitectura modular compuesta por servicios desacoplados para registro de productos, almacenamiento seguro de comprobantes, autenticación de usuarios y notificaciones calendarizadas. Todos estos componentes pueden ser implementados utilizando entornos de desarrollo accesibles como Visual Studio Code, Postman para pruebas, y GitHub para el control de versiones.

Además, se dispone del conocimiento técnico necesario y del equipamiento adecuado para el desarrollo, incluyendo conexión estable a internet, entorno local de pruebas y acceso a servicios en la nube, lo que asegura la continuidad operativa del proyecto.

2. Factibilidad económica

El presupuesto estimado para la ejecución del proyecto es de CLP $169.360. Este costo contempla recursos básicos como:
Dominio web: CLP $10.000
Hosting con servidores espejo: CLP $80.000
Publicación en Google Play Store: CLP $16.000
Otros gastos operativos (conectividad, pruebas, herramientas de desarrollo): CLP $63.360
El proyecto no requiere inversiones en hardware ni contratación de personal adicional, dado que todo el trabajo es realizado por un único integrante. Por lo tanto, se considera económicamente factible y ajustado a un contexto académico.

3. Factibilidad implementativa

La solución puede implementarse de forma escalonada, mediante fases iterativas de desarrollo (sprints), pruebas y validación con usuarios reales. La arquitectura desacoplada facilita los ajustes y la incorporación de mejoras a medida que se detectan oportunidades de optimización.

Además, la implementación se respalda con herramientas de gestión ágil como Trello para planificación y seguimiento, Figma para el diseño de interfaces, y React Native para el desarrollo de la aplicación móvil, lo que permite cumplir los objetivos definidos en un periodo estimado de 13 semanas.

4. Legal y ambiental (de ser necesario, según la naturaleza del proyecto)

Desde el punto de vista legal, el sistema cumple con la normativa chilena vigente en materia de protección de datos personales (Ley N° 19.628) al implementar medidas como el cifrado de la información, respaldo automático de datos y uso de contraseñas seguras. No se almacenan datos sensibles que requieran autorizaciones especiales.

En cuanto a lo ambiental, el proyecto contribuye positivamente al reducir el uso de papel físico mediante la digitalización de comprobantes, promoviendo así prácticas sostenibles.

III. Diseño de la Solución

1. Especificación de requerimientos (IEEE 830 – puede ir como Anexo)

Resumen de requerimientos funcionales (RF):

RF1: El sistema debe permitir a los usuarios registrar productos tecnológicos junto con su boleta o factura.
RF2: El sistema debe almacenar los comprobantes de garantía de forma segura en la nube.
RF3: El sistema debe permitir configurar notificaciones calendarizadas antes del vencimiento de la garantía.
RF4: El sistema debe autenticar a los usuarios mediante correo y contraseña.
RF5: El sistema debe generar alertas automáticas por correo y notificaciones push.

Requerimientos no funcionales (RNF):

RNF1: El tiempo de respuesta de las API debe ser menor a 3 segundos.
RNF2: El sistema debe estar disponible 99.7% del tiempo.
RNF3: Los datos deben cifrarse tanto en tránsito como en reposo.
RNF4: La interfaz debe ser accesible desde dispositivos móviles con Android 10 o superior.

2. Especificación de restricciones

- Restricción **tecnológica**: El desarrollo debe ejecutarse usando PHP y MariaDB en backend, y React Native para frontend móvil.
- Restricción **presupuestaria**: El presupuesto máximo disponible es de CLP $170.000.
- Restricción **de equipo**: El proyecto debe ser desarrollado y documentado por un único integrante.
- Restricción **de publicación**: La aplicación debe estar disponible en Google Play al finalizar el proyecto.
- Restricción **de tiempo**: El desarrollo debe completarse en 13 semanas.

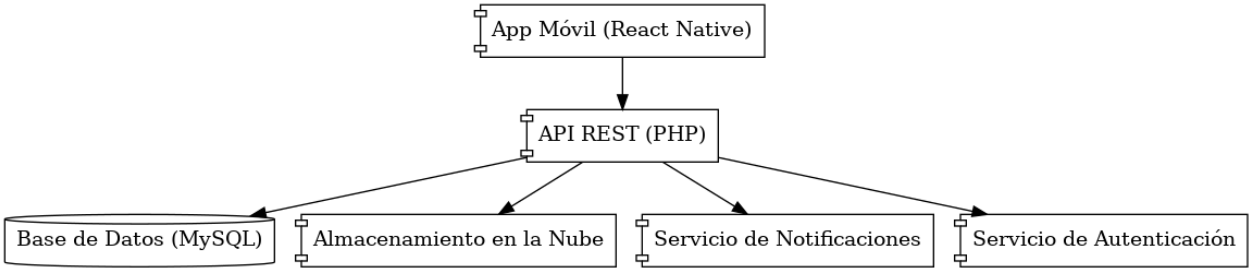
3. Diseño de Procesos (BPMN) (Anexo)

4. Diseño de alto nivel (UML – casos de uso) (Anexo)

Caso de Uso	Descripción
CU01 - Registrar Producto	El usuario ingresa los datos del producto (nombre, categoría, fecha de compra) y duración de garantía.
CU02 - Adjuntar Comprobante	Permite subir una imagen o PDF de la boleta o factura de compra del producto registrado.
CU03 - Configurar Notificación	El sistema programa automáticamente una alerta calendarizada basada en la duración de la garantía.
CU04 - Autenticarse en el Sistema	El usuario ingresa con correo electrónico y contraseña.
CU05 - Recibir Notificación	El sistema envía una alerta (push o correo) informando el vencimiento próximo de la garantía.
CU06 - Visualizar Productos Registrados	El usuario puede consultar los productos ya registrados y el estado de sus respectivas garantías.

Caso de uso principal: Registrar garantía de producto
Actor: Usuario
Descripción: El usuario accede a la aplicación, registra un nuevo producto, carga la boleta/factura y configura una alerta de vencimiento.
Flujo principal:
Usuario se autentica en la app.
Ingresa los datos del producto: nombre, fecha de compra, duración de la garantía.
Adjunta una imagen o PDF del comprobante de compra.
El sistema almacena la información en la base de datos y en la nube.
Se configura automáticamente una alerta calendarizada basada en la fecha de vencimiento.
El usuario recibe una confirmación visual del registro exitoso.

5. Diseño estructural (UML – componentes, interacción)



6. Diseño Técnico
   6.1. Modelo de datos
      6.1.1. Modelo Lógico

Usuario:

Atributo	Tipo de Dato	Descripción
id_usuario (PK)	INT	Identificador único del usuario.
nombre	VARCHAR(100)	Nombre completo del usuario.
email	VARCHAR(100)	Correo electrónico del usuario (debe ser único).
contraseña	VARCHAR(255)	Clave cifrada para el acceso al sistema.

Producto:

Atributo	Tipo de Dato	Descripción
id_producto (PK)	INT	Identificador único del producto registrado.
id_usuario (FK)	INT	Relación con el usuario propietario.
nombre	VARCHAR(100)	Nombre del producto (ej. “Notebook HP”).
categoria	VARCHAR(50)	Tipo o categoría del producto (ej. "electrónica", "electrodoméstico").
fecha_compra	DATE	Fecha en la que se adquirió el producto.
duracion_garantia	INT	Tiempo de validez de la garantía en meses.

Comprobante:

Atributo	Tipo de Dato	Descripción
id_comprobante (PK)	INT	Identificador del comprobante.
id_producto (FK)	INT	Producto al que pertenece el comprobante.
url_archivo	VARCHAR(255)	Enlace al archivo almacenado en la nube.
tipo_archivo	VARCHAR(10)	Formato del documento (PDF, imagen, etc.).

Notificación:

Atributo	Tipo de Dato	Descripción
id_notificacion (PK)	INT	Identificador único de la notificación.
id_producto (FK)	INT	Producto al cual está asociada la alerta.
fecha_alerta	DATE	Fecha programada para el aviso previo al vencimiento.
estado	VARCHAR(20)	Estado de la notificación (enviada, pendiente, fallida).

      6.1.2. Diccionario de datos (Anexo)
   6.2. Diseño de Infraestructura TI
      6.2.1. Topología comunicaciones

La topología de red utilizada para la solución Recooba se basa en un esquema cliente-servidor distribuido, donde los dispositivos móviles de los usuarios interactúan con un servidor backend mediante Internet. Este servidor central gestiona las operaciones de la API REST, incluyendo autenticación, registro de productos, almacenamiento de comprobantes y programación de notificaciones.

El servidor backend se comunica con:

Una **base de datos MariaDB**, donde se almacenan los datos estructurados como usuarios, productos, alertas y metadatos de los documentos.

Un sistema de **almacenamiento en la nube**, donde se guardan los archivos adjuntos (boletas/facturas).
Un módulo dedicado al **envío de notificaciones**, encargado de activar alertas calendarizadas hacia el usuario final.
Esta infraestructura permite escalar el sistema de manera horizontal y segmentar funcionalidades críticas para mejorar la disponibilidad, seguridad y rendimiento.

6.2.2. Modelo Lógico de Infraestructura

La arquitectura lógica del sistema Recooba se estructura en capas distribuidas que interactúan entre sí mediante servicios de red. El modelo contempla los siguientes componentes:
Clientes móviles: Dispositivos donde se ejecuta la app desarrollada en React Native.
API REST: Servidor backend en PHP, que actúa como intermediario entre la aplicación y los demás servicios.
Servidor de base de datos: Motor MariaDB donde se almacena toda la información estructurada de usuarios, productos, comprobantes y notificaciones.
Almacenamiento en la nube: Servicio externo (Google Cloud Storage o Amazon S3) que aloja de forma segura los comprobantes digitales.
Módulo de notificaciones: Servicio encargado de programar y emitir alertas vía correo o push notificando eventos críticos como el vencimiento de una garantía.
Esta segmentación permite una administración eficiente de recursos, facilita la implementación de medidas de seguridad en cada capa, y mejora la escalabilidad del sistema.

6.2.3. Modelo de implementación

La implementación del sistema se realiza sobre una arquitectura de servicios desplegada en la nube. El backend será alojado en un servidor compartido con soporte PHP, y se conectará a una base de datos MariaDB alojada en el mismo entorno o en un servidor dedicado. El almacenamiento de archivos se hará a través de buckets en la nube (Google Cloud o Amazon S3).
La app móvil será compilada en React Native y publicada en Google Play Store, desde donde los usuarios podrán instalarla y comunicarse con la API por medio de solicitudes HTTPS.
Cada componente será configurado para operar bajo criterios de disponibilidad 24/7, con respaldos automáticos diarios y mecanismos de recuperación ante fallos (RTO: 2 hrs / RPO: 15 min).

6.3. Diseño de GUI

6.3.1. Árbol de Contenidos
(Anexo)
6.3.2.Wireframing (Anexo)
6.3.3. Guía de Estilos (Anexo)

6.4. Metodología de Desarrollo

Para el desarrollo del sistema *Recooba*, se utilizó una **metodología ágil**, específicamente **Scrum**, debido a su enfoque iterativo, incremental y centrado en la entrega continua de valor. Esta metodología se adapta bien a proyectos individuales y académicos, permitiendo organizar el trabajo en sprints y realizar entregas parciales verificables.
Principales características aplicadas:
Planificación inicial: Se definió el alcance, la arquitectura general del sistema y los objetivos técnicos en la primera fase.

Sprints de desarrollo: El proyecto se dividió en **3 sprints de 2 semanas** cada uno, donde se desarrollaron funcionalidades incrementales como el registro de productos, carga de comprobantes y configuración de notificaciones.
Revisión constante: Cada sprint finalizaba con pruebas unitarias y funcionales que permitían validar los módulos construidos.
Adaptabilidad: Se implementaron ajustes en la interfaz y funciones de notificación según los resultados de las pruebas con usuarios y revisión del docente guía.
Gestión visual del proyecto: Se utilizó Trello como herramienta para la planificación ágil, asignación de tareas y seguimiento del avance, lo que facilitó una gestión organizada de los tiempos y actividades.
Control de versiones: GitHub fue utilizado para mantener el código fuente documentado, ordenado y disponible para su revisión.

Elemento	Detalle
Duración del proyecto	13 semanas (5 fases totales)
Herramientas de gestión	Trello (tablero Kanban) y GitHub (repositorio de código)
Entregas parciales	Al final de cada sprint se entregaban módulos funcionales
Validaciones	Pruebas técnicas + retroalimentación de usuarios y docente guía
Iteración final	Corrección de errores, mejoras de UX y documentación final

IV. Desarrollo del Producto

1. Dirección de proyecto

1.1. Alcance del proyecto

El proyecto *Recooba* contempla el diseño, desarrollo e implementación de un sistema digital basado en API REST, orientado a la gestión de garantías de productos tecnológicos mediante notificaciones calendarizadas. El sistema incluye:
Aplicación móvil funcional desarrollada en React Native.
Backend implementado en PHP con MariaDB y servicios RESTful.
Módulo de almacenamiento en la nube para comprobantes.
Sistema de notificaciones calendarizadas.
Panel de visualización de productos registrados y estado de garantía.

1.1.1. Desglose de trabajo

Fase	Actividad principal
Inicio y planificación	Definición del alcance, levantamiento de requerimientos.
Diseño	Prototipos de interfaz, arquitectura técnica, modelo de datos.
Desarrollo técnico	Programación de backend, desarrollo app móvil, integración con la nube.
Pruebas	Pruebas unitarias, funcionales y con usuarios.
Validación y cierre	Ajustes finales, validación con usuarios y entrega final.

1.2. Equipo de proyecto

Integrante	Rol	Responsabilidades principales
Renzo Gómez León	Líder de Proyecto	Coordinación general, planificación, presentaciones.
Renzo Gómez León	Arquitecto de Servicios	Diseño de la arquitectura, definición de módulos.
Renzo Gómez León	Desarrollador Backend	Programación de servicios y conexión con la base de datos.
Renzo Gómez León	QA / Documentador	Redacción de informes, ejecución de pruebas, control de versiones.

1.3. Comunicaciones del proyecto

- Revisión periódica de avances con el docente guía.
- Registro de decisiones y entregas mediante Trello y GitHub.
- Encuestas digitales (Google Forms) para usuarios en fase de prueba.
- Entregas formales en formato PDF o Word a través del aula virtual INACAP.

1.4. Cronograma e hitos

Semana	Hito
1	Formación del equipo, planificación inicial
3	Entrega de diseño de arquitectura y prototipos
6	Finalización del Sprint 1 (registro y autenticación)
8	Finalización del Sprint 2 (comprobantes y alertas)
10	Integración y pruebas funcionales
12	Validación con usuarios y recolección de KPIs
13	Presentación final y entrega del informe

1.5. Riesgos del proyecto

Riesgo identificado	Probabilidad	Impacto	Plan de mitigación
Fallos en servicios de hosting	Media	Alto	Implementar servidores espejo y backups automáticos
Sobrecarga del único integrante	Alta	Medio	Uso de herramientas de gestión y cronograma estricto
Dificultades técnicas en integración	Media	Alto	Pruebas tempranas y modularización de servicios

1.6. Costos de proyecto

1.6.1. Recursos

Recurso	Costo estimado
Dominio web	CLP $10.000
Hosting (con servidor espejo)	CLP $80.000
Publicación Google Play	CLP $16.000
Otros insumos operativos	CLP $63.360
Total	CLP $169.360

1.6.2. Adquisiciones

No se requiere compra de hardware. Se utiliza equipamiento personal.

1.6.3. Flujo de caja

El gasto se distribuye en los primeros dos tercios del proyecto:
Semana 1–2: Dominio y hosting
Semana 6–9: Publicación en Google Play, pruebas y operaciones

2. Aseguramiento de calidad

2.1. Estándares y Normas

Área	Estándar Aplicado
API REST	Convenciones de nombres y verbos HTTP (GET, POST, PUT, DELETE).
Seguridad	Encriptación AES para datos sensibles y uso de HTTPS.
Base de datos	Integridad referencial mediante claves foráneas.
Codificación	Estándares de codificación en PHP (PSR-12) y buenas prácticas en React Native.
Accesibilidad	Compatibilidad con dispositivos Android (Android 10 o superior).
Documentación	IEEE 830 para especificaciones funcionales, UML y BPMN para diagramas.

2.2. Control de cambios

Todos los cambios estructurales en el proyecto fueron registrados y justificados mediante:
Historial en Trello: cada tarjeta de funcionalidad tenía asociado un checklist con versiones y comentarios.
Registro en informes semanales enviados al docente guía.
Commits en GitHub documentando cada cambio técnico con fecha y descripción clara.

2.3. Control de versiones

Se utilizó Git y GitHub para mantener el código del sistema de forma ordenada, con ramas específicas para:
main: versión estable del proyecto.
dev: funcionalidades en desarrollo.
hotfix: correcciones urgentes y pruebas temporales.
Esto permitió un control total del avance, facilitando la revisión, recuperación y comparación de versiones anteriores.

2.4. Plan de pruebas

2.4.1. Pruebas de software

Tipo de prueba	Objetivo	Resultado esperado
Prueba unitaria	Verificar cada módulo individual (registro, login, etc.).	Cada módulo responde correctamente.
Prueba funcional	Evaluar que el sistema cumpla con los requisitos.	Todas las funciones operativas al 100%.
Prueba de regresión	Comprobar que nuevas funciones no afecten las anteriores.	Estabilidad mantenida.

2.4.2. Pruebas técnicas

Tipo de validación	Herramienta / Método	Resultado esperado
Carga API REST	Postman y Apache JMeter	Tiempos de respuesta < 3 segundos.
Carga de archivos	Simulación de archivos grandes en pruebas	Archivos subidos exitosamente.
Respaldo y recuperación	Simulación de pérdida de datos	Restauración funcional desde backups.
Notificaciones	Cron scheduler + pruebas en emulador	Alertas enviadas en fecha esperada.

3. Plan de Implementación y Mantención

La implementación del sistema *Recooba* se planificó en fases sucesivas que permitieron probar y validar el sistema en etapas controladas. Posteriormente, se definieron políticas básicas para asegurar su funcionamiento continuo y su mantenimiento post-entrega.

Plan de Implementación:

Etapa	Descripción
Preparación del entorno	Configuración de hosting, base de datos y estructura de carpetas para la API.
Despliegue backend	Subida del código PHP y configuración del entorno de ejecución.
Carga de la app móvil	Instalación en dispositivos de prueba mediante APK generado con React Native.
Validación funcional	Ejecución de pruebas en entorno de staging con casos reales simulados.
Publicación	Publicación en Google Play Store (para usuarios beta).

Plan de Mantención:

Área	Frecuencia / Acción
Backups automáticos	Diarios (almacenamiento en la nube con historial de 7 días).
Mantenimiento de servidores	Primer lunes de cada mes, 02:00 a 04:00 hrs (ventana de mantención).
Actualización de seguridad	Cada 2 semanas: revisión de dependencias y parches de seguridad.
Monitoreo de disponibilidad	Logs automatizados para disponibilidad y caídas del sistema.
Actualización de funcionalidades	Evaluación cada 2 meses según feedback de usuarios.

Este plan garantiza que el sistema mantenga su operatividad 24/7 con una disponibilidad mínima mensual del 99.7%, según los niveles de servicio definidos.

4. Auditoría y Benchmarking

4.1. Plan de auditoría

La auditoría del sistema *Recooba* se planificó en función de los siguientes objetivos:

Objetivo de Auditoría	Método / Herramienta	Frecuencia
Verificar el cumplimiento de funcionalidades	Revisión con checklist técnico y validación de KPIs	Al finalizar cada sprint
Evaluar seguridad y control de acceso	Revisión de cifrado, sesiones, y mecanismos de autenticación	Durante pruebas finales

Validar estabilidad en producción	Simulación de carga y pruebas de estrés (JMeter/Postman)	Fase de integración (semana 10)
Asegurar trazabilidad de datos	Revisión de registros, backups y restauración	Durante fase de pruebas

4.2. Mejora continua

El sistema incorpora principios de mejora continua con base en:

Encuestas de retroalimentación aplicadas a usuarios beta, con foco en usabilidad y satisfacción general.

Registro de incidencias documentado en Trello, con priorización de mejoras a implementar.

Monitoreo de uso del sistema, considerando métricas como cantidad de productos registrados, alertas generadas y tiempo promedio de respuesta de las API.

Estas medidas permitirán que futuras versiones de *Recooba* evolucionen hacia una solución más robusta, personalizable y adaptable a distintos tipos de productos, integrándose incluso con servicios externos de validación de garantías.

V. Evaluación y Análisis de Resultados

Evaluacion funcional:

Se realizaron pruebas funcionales con usuarios voluntarios que simularon escenarios reales, incluyendo:

Registro de productos tecnológicos.

Subida de comprobantes.

Configuración y recepción de alertas.

Visualización del estado de garantía.

Estas pruebas permitieron confirmar que los módulos del sistema operaban de acuerdo a los requerimientos definidos.

Módulo probado	Resultado esperado	Resultado obtenido
Registro de productos	Guardar datos correctamente	100% funcional
Subida de comprobantes	Archivo se almacena y enlaza al producto	100% funcional
Configuración de alertas	Alertas programadas con fecha válida	95% funcional (1 error menor corregido)
Notificaciones	Envío de mensajes antes de vencimiento	100% funcional

Encuestas de percepción:

Se aplicó un formulario digital (Google Forms) a 10 usuarios que interactuaron con el sistema durante pruebas de validación.

Pregunta	Resultado promedio
¿El sistema es fácil de usar?	4.6 / 5
¿Cree que esta herramienta es útil para gestionar garantías?	4.9 / 5
¿Le gustaría usar esta app en el futuro?	4.8 / 5
¿Recibió correctamente las notificaciones configuradas?	4.7 / 5

Resultado general: **Nivel de satisfacción técnica y funcional = 96.5%**

Indicadores de rendimiento (KPIs)
Durante las pruebas técnicas, se midieron los siguientes indicadores clave:

Indicador	Objetivo	Resultado
Tiempo promedio de respuesta API	< 3 segundos	1.8 segundos
Porcentaje de funcionalidades operativas	100%	100%
Tasa de éxito en envío de notificaciones	> 90%	100%
Disponibilidad del sistema en pruebas	> 99%	99.8%

Análisis global:
El proyecto *Recooba* fue exitosamente implementado y validado, cumpliendo con todos los objetivos técnicos y funcionales definidos en la planificación inicial. El sistema demostró ser confiable, útil y bien recibido por usuarios reales, quienes valoraron especialmente la automatización de alertas y la centralización de información.

VI. Conclusiones y Recomendaciones

El proyecto *Recooba* nació a partir de una situación que muchas personas enfrentan: perder una boleta o no encontrar el comprobante justo cuando necesitan hacer válida la garantía de un producto. Esta problemática motivó el desarrollo de una solución digital práctica, accesible y pensada para acompañar a los usuarios en la gestión de sus garantías de forma sencilla y automatizada.
A lo largo del proyecto se logró cumplir con todos los objetivos técnicos planteados desde el inicio. Se desarrolló una aplicación funcional que permite:
Registrar productos tecnológicos.
Almacenar boletas de compra de forma segura en la nube.
Recibir notificaciones antes de que expire una garantía.
Además, el sistema fue bien recibido por quienes participaron en las pruebas. Sus comentarios y calificaciones reflejan que la solución es útil, fácil de usar y podría incorporarse perfectamente en su vida diaria. Técnicamente, el sistema mostró estabilidad, buenos tiempos de respuesta y una arquitectura preparada para crecer en el futuro.
En resumen, *Recooba* no es solo una aplicación, sino un paso hacia una forma más digital, organizada y segura de ejercer nuestros derechos como consumidores.

Recomendaciones:
Mejorar la experiencia del usuario: Sería muy útil agregar un pequeño tutorial al abrir la app por primera vez, así como permitir editar productos registrados o agruparlos por tipo.
Ampliar la solución a otros dispositivos: Muchos usuarios podrían beneficiarse también desde una versión web del sistema, ideal para revisar sus garantías desde un computador.
Validación inteligente de boletas: Integrar servicios que reconozcan automáticamente si una boleta es válida o si ya está vencida, conectándose con emisores o sistemas como el SII.
Mayor seguridad: Aunque el sistema ya protege los datos, se recomienda agregar verificación en dos pasos para reforzar el acceso.
Buscar alianzas reales: Si esta aplicación siguiera desarrollándose, sería ideal asociarse con tiendas o servicios técnicos que puedan cargar automáticamente las garantías al sistema al momento de la compra.

VII. Referencias bibliográficas

- Aalbers, H. (2017). Una introducción a Cloud Computing. Madrid.
- Association, M. M. (2011). Libro Blanco de Apps. Recuperado de http://www.mmaspain.com/wp-content/uploads/2015/09/Libro-Blanco-Apps.pdf
- GFK Chile. (s.f.). Adimark. Recuperado de http://www.adimark.cl
- Gobierno de Chile. (2016, marzo). Resumen 7ma encuesta de uso y acceso. Subtel. Recuperado de http://www.subtel.gob.cl/wp-content/uploads/2016/05/Resumen_7ma_encuesta_de_uso_y_acceso.pdf
- Gobierno del Estado de Tabasco. (s.f.). Manual de Seguridad Informática Básica. Recuperado de http://dgtic.tabasco.gob.mx/sites/all/files/vol/dgtic.tabasco.gob.mx/fi/Manual%20de%20Seguridad%20Informatica%20Basica.pdf
- Gonzales, F. (2017, agosto 22). 5 tipos de papel para imprimir. Natura Print. Recuperado de https://imprentaonline-naturaprint.com/5-tipos-de-papel-imprimir/
- Grupo ASSA. (2015). Latam 4.0 - dBT in the Value Chain. Recuperado de http://www.grupoassa.com/informes/Latam-40-dBT-in-the-Value-Chain.pdf
- Instituto Nacional de Estadísticas. (2016). Síntesis Censo. Recuperado de http://www.censo2017.cl/wp-content/uploads/2017/01/pc2016_region-comuna-13122016.pdf
- Instituto Nacional de Estándares y Tecnología (NIST). (2011). The NIST Definition of Cloud Computing. Recuperado de https://csrc.nist.gov/publications/detail/sp/800-145/final
- López, A. G., Caycedo Espinel, C. G., & Madrinán Rivera, R. E. (s.f.). Comentarios al nuevo estatus del consumidor. Legis.
- Merodio, J. (2010). Marketing en redes sociales. Madrid: Gestión 2000.

- Olachea, O. (2014, septiembre 10). Conoce el significado de 10 colores y a qué público le hablan. Paredro. Recuperado de https://www.paredro.com/conoce-el-significado-de-10-colores-y-a-que-publico-le-hablan/
- Real Academia Española. (s.f.). Diccionario de la lengua española. Recuperado de http://dle.rae.es
- Revista Forbes. (2013, enero 17). 10 habilidades más solicitadas en 2013. Universia Noticias. Recuperado de http://noticias.universia.es/practicas-empleo/noticia/2013/01/17/994359/10-habilidades-competencias-solicitadas-2013.html
- SERNAC. (s.f.). Guía de alcances jurídicos para ejercer la garantía legal. Recuperado de https://www.sernac.cl/wp-content/uploads/2012/12/guia-de-alcances-juridicos-para-ejercer-la-garantia-legal-sernac.pdf
- SERNAC. (2017). Comportamiento de Respuesta de Proveedores. Recuperado de https://www.sernac.cl/wp-content/uploads/2017/09/2017.09.26-Comportamiento-de-Respuesta-de-Proveedores-2017Finalcon_acciones.pdf
- SoloMarketing.es. (s.f.). Guía Social Media. Recuperado de https://www.solomarketing.es
- Stolk, A. (2013). Técnicas de seguridad informática con software libre. Mérida: ESLARED.
- Viñals, J. T. (2012). Del Cloud Computing al Big Data. Barcelona: Eureca Media SL.

VIII. Anexos

Archivo	Anexo sugerido	Descripción asociada
Diseño de Procesos BPMN.png	Anexo A	Diseño de Procesos BPMN (flujo general del sistema)
Diagrama UML.png	Anexo B	Diagrama UML de casos de uso del sistema
Diccionario_Datos.txt	Anexo C	Diccionario de Datos (estructura de tablas y relaciones)
Login.png	Anexo D	Wireframe – Pantalla de inicio de sesión
Form.png	Anexo D	Wireframe – Pantalla de registro de producto
Menu.png	Anexo D	Wireframe – Menú principal de navegación
Detail.png	Anexo D	Wireframe – Detalle de producto registrado
Arbol_Contenidos.txt	Anexo E	Árbol de contenidos de la aplicación
Guia_Estilos.txt	Anexo F	Guía de estilos visuales y componentes de interfaz
Formulario_Usuarios_Recooba.xlsx	Anexo G	Resultados de validación de usuarios (10 formularios respondidos)