

UNIVERSIDAD DE TARAPACÁ

INGENIERÍA CIVIL EN COMPUTACIÓN E INFORMÁTICA
ARICA – CHILE



Departamento de Ingeniería en Computación e Informática



Proyecto IV “Informe de Avance II”

Autores: José Díaz
Cristián Fritis

Asignatura: Proyecto IV

Profesor: Diego Aracena

Arica, Chile

2022



Tabla de Contenido

1. Introducción.....	4
2. Objetivos.....	5
2.1. Objetivo General.....	5
2.2. Objetivos Específicos.....	5
3. Descripción de la empresa Kuvemar.....	6
3.1. Visión de la empresa.....	6
3.2. Misión de la empresa.....	6
3.3. Valores.....	6
3.4. Historia.....	6
3.5. Organigrama de Kuvemar.....	6
3.6. Servicios ofrecidos.....	7
4. Definición del problema.....	8
4.1. Contexto.....	8
4.2. Problema.....	8
4.3. Solución.....	8
5. Planificación general.....	10
5.1. Requerimientos.....	10
5.1.1. Requerimientos funcionales.....	10
5.1.2. Requerimientos no funcionales.....	11
5.1.3. Acta de acuerdo.....	11
5.2. Carta Gantt.....	12
5.3. Modelo de contexto.....	13
5.3.1. Diagrama de comunicación subsistemas.....	13
5.3.2. Diagrama de interacción entre subsistemas.....	14
5.4. Gestión del proyecto.....	15
5.4.1. Herramientas a utilizar.....	15
5.4.2. Framework Angular.....	16
5.4.3. Framework NodeJS.....	18
5.5. Modelamiento de la aplicación.....	19
5.5.1. Identificación del producto/servicio.....	19
5.5.2. Despliegue del estado de un producto/servicio.....	20
5.6. Herramientas a utilizar.....	21
5.7. Arquitectura del sistema.....	22



5.8. Diagramas BPMN.....	22
5.8.1. Despliegue estado de un producto/servicio.....	22
5.8.2. Generar PDF.....	23
5.8.3. Notificación automática.....	24
5.9. Casos de Usos de Sistema.....	25
5.9.1. Descripción casos de Usos de Sistema.....	25
5.10. Diagramas de Secuencia.....	28
5.10.1. Diagrama de secuencia “Despliegue producto/servicio”.....	28
5.10.2. Diagrama de secuencia “Notificaciones automáticas”.....	29
5.10.3. Diagrama de secuencia “Creación de PDF”.....	30
5.11. Base de datos.....	31
5.12. Repositorio.....	32
6. Conclusiones.....	33



Índice de figuras

Figura 1, Organigrama Kuvemar.	6
Figura 2, Carta Gantt del proyecto.....	12
Figura 3, Diagrama de contexto.....	13
Figura 4, Diagrama de comunicación entre subsistemas.	13
Figura 5, Diagrama de Interacción.....	14
Figura 6, Estructura de carpetas Angular.	17
Figura 7, Logo NodeJs.....	18
Figura 8, Identificación Producto/servicio.	19
Figura 9, Dashboard del estado de un producto/servicio.	20
Figura 10, Arquitectura del software a utilizar para el proyecto.	22
Figura 11, Diagrama BPMN “Despliegue estado de un producto/servicio”.	22
Figura 12, Diagrama BPMN “Generar PDF”.	23
Figura 13, Diagrama BPMN “Notificación a usuarios.	24
Figura 14, Casos de uso del sistema.....	25
Figura 15, Diagrama de secuencia Nivel 0 - Despliegue de estado de producto/servicio.	28
Figura 16, Diagrama de secuencia Nivel 0 - Notificación por Correo del estado del servicio.	29
Figura 17, Diagrama de secuencia Nivel 0 - Descargar PDF.	30
Figura 18, Sección base de datos Kuvemar.	31
Figura 19, GitHub desktop del Proyecto.	32

Índice de tablas

Tabla 1, Requerimientos funcionales.....	10
Tabla 2, Requerimientos no funcionales.....	11
Tabla 3, Caso de uso "Descargar PDF".	25
Tabla 4, Caso de uso "Consultar estado del producto/servicio".	26
Tabla 5, Caso de uso "Notificar cliente".	27



1. Introducción.

La organización es lo fundamental para llegar a implementar de la mejor forma cualquier idea que se tenga propuesta. Esto mismo, se puede llevar a la realización de un proyecto de gran envergadura, teniendo una planificación en donde los pasos si son varios, pero llevan de manera general a la mejor solución posible.

Por ende, en el presente informe se presentará cada avance, el cual se haya realizado en el transcurso de los meses de desarrollo. Integrando entre ellos el modelo de contexto, casos de uso, requerimientos formales de la aplicación, diagramas BPMN, diagramas de secuencia, y base de datos con la cual se trabajó para llevar a cabo la implementación de los requisitos a mencionar.



2. Objetivos.

2.1. Objetivo General.

Presentar toda la documentación e información necesaria para lograr entender el desarrollo del proyecto con nombre “Módulos para la presentación de estados de proyectos a clientes”.

2.2. Objetivos Específicos.

- Investigar y caracterizar el ambiente de trabajo.
- Analizar y diseñar las interacciones entre los distintos módulos a implementar.
- Desarrollar el sistema web gestionado de proyectos.
- Realizar pruebas y posibles correcciones del sistema.
- Comprobar los resultados y realizar un análisis de las partes que restan.



3. Descripción de la empresa Kuvemar.

3.1. Visión de la empresa.

Ser el principal proveedor de servicios de desarrollo tecnológico a nivel nacional con una cultura de innovación constante, reconocida por su profesionalismo, confianza y adaptabilidad en potenciar la propuesta de valor de cada empresa cliente.

3.2. Misión de la empresa.

Apoyar a las empresas a través de herramientas estratégicas y tecnológicas que potencien y logren generar un impacto productivo en sus procesos de negocios.

3.3. Valores.

La esencia de Kuvemar se basa en la confianza, compromiso, adaptabilidad, profesionalismo, respeto y trabajo en equipo.

3.4. Historia.

La empresa Kuvemar se establece en el año 2017 en la ciudad de Arica, dentro de la región de Arica y Parinacota. Inicialmente, la empresa se compuso por un equipo de 7 desarrolladores, donde actualmente, el equipo cambio su cantidad de desarrolladores. Actualmente, el equipo de trabajo de Kuvemar se encuentra compuesto por un total de 4 desarrolladores, donde cada uno se encuentra especializado en un área específica de la programación.

3.5. Organigrama de Kuvemar.

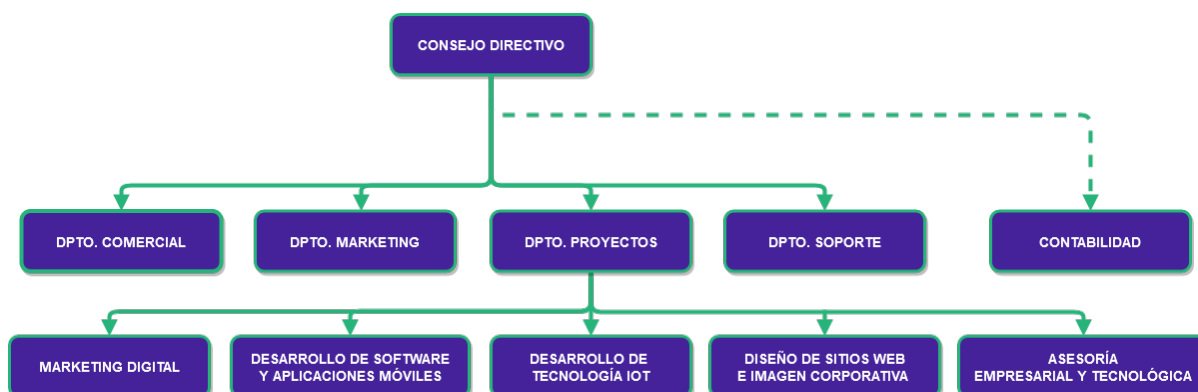


Figura 1, Organigrama Kuvemar.



3.6. Servicios ofrecidos.

La empresa Kuvemar ofrece una variedad de servicios orientados al desarrollo de aplicaciones software y de creación de páginas webs para el cliente, tales como:

- Desarrollo de Software.
- Aplicaciones Móviles.
- Páginas Web.
- Tiendas eCommerce.
- Asesorías Tecnológicas.
- SaaS.



4. Definición del problema.

4.1. Contexto.

La empresa Kuvemar establecida en la región de Arica y Parinacota, es una empresa enfocada al desarrollo tecnológico de la ciudad de Arica. La empresa nace con el objetivo de potenciar a las micro, pequeñas y medianas empresas a través de distintas herramientas tecnológicas para automatizar los procesos de negocios, la atracción de potenciales clientes, mejorar la experiencia de compra y posicionamiento de productos/servicios en plataformas digitales. Debido al incremento de clientes, la empresa Kuvemar ha notado la recurrencia existente de parte de los mismos en consultar cómo va el progreso de productos/servicios que están en desarrollo. Aunque esto no sea un problema grave, se pierde cierta cantidad de tiempo diariamente en dar respuestas a estas solicitudes. Por ende, se ha originado la idea de realizar un sistema que implemente una solución a este problema en concreto. Con el cual el cliente podrá realizar estas solicitudes, y obtener respuesta de manera inmediata mediante la automatización de este servicio.

4.2. Problema.

El problema consiste en la demanda de consultas que realizan los clientes, refiriéndose a los estados de avances de un producto/servicio, ya sea de software, app. Móvil, página o tiendas web, entre otros. La cual genera retrasos de respuesta al cliente, por esto, se busca digitalizar el proceso con la finalidad de mantener informado al cliente sobre nuevos avances de su producto/servicio. Además, de poder ofrecer otras implementaciones como un resumen de las actividades que se encuentren anidadas a un producto/servicio mediante un reporte realizado en un archivo con extensión .pdf, y un sistema automatizado, el cual, con una periodicidad semanal, debe realizar envío a cada cliente por separado, del estado de avance de sus productos/servicios, los cuales se encuentren asociados a su persona.

4.3. Solución.

Se tiene como solución desarrollar un sistema de seguimiento de etapas del producto/servicio del cliente, un sistema automático para brindarle notificaciones al mismo y finalmente un módulo el cual permita generar un resumen del estado del producto/servicio en un archivo con extensión .pdf.



Estas implementaciones tendrán como función desplegar todo el estado relevante del producto/servicio al cliente, añadir la facilidad de poder enviar notificaciones automáticas al mismo cuando haya avances del producto/servicio, entregar la posibilidad de generar un resumen del estado del producto/servicio en un archivo .pdf para que el cliente pueda descargar y tener a la mano una copia del desarrollo de su proyecto y finalmente que permita informar mediante notificaciones al cliente en el caso de que ocurra algún cambio en los recursos necesarios para el desarrollo del proyecto.



5. Planificación general.

5.1. Requerimientos.

A continuación, se mostrarán los requerimientos funcionales y no funcionales solicitados por el cliente, la cual tiene que cumplir el proyecto para su posterior aprobación.

5.1.1. Requerimientos funcionales.

Tabla 1, Requerimientos funcionales.

Número	Requisito Funcional	Descripción
RF-1	Consulta del estado de un producto/servicio.	El sistema debe permitir al cliente ingresar un número de serie con el cual se identificará su producto/servicio.
RF-2	Despliegue del estado de un producto/servicio.	El sistema debe entregar el estado de avance de un producto/servicio, cuando se le consulta por parte del cliente. El estado de avance debe ser identificado mediante un número de serie (desplegando las actividades completadas/no completadas las cuales están anidadas al producto/servicio).
RF-3	Notificación automática del estado de un producto/servicio.	El sistema automáticamente con una periodicidad semanal debe realizar envío a cada cliente por separado, el estado de avance de sus productos/servicios, los cuales se encuentren asociados a su persona (desplegando las actividades completadas/no completadas las cuales están anidadas al producto/servicio).
RF-4	Generar PDF del estado de un producto/servicio.	El sistema tendrá la funcionalidad de generar un archivo .pdf el cual contenga un resumen de las actividades del producto/servicio por el cual se esté consultando (anidando las actividades completadas/no completadas las cuales están anidadas al producto/servicio).
RF-5	Contenido del estado de un producto/servicio.	El contenido que deberá tener el estado del producto/servicio al momento de ser solicitado por parte del cliente, o al momento de ser utilizado para la notificación automática, mediante el uso de correos electrónicos, debe ser el de un listado, el cual debe estar compuesto por las actividades completadas, no completadas, y en desarrollo, donde cada una de estas debe estar identificada por un color, junto a su respectiva fecha de inicio y finalización por cada actividad. Para esto, se realizará uso de los diferentes colores: <ul style="list-style-type: none">- Actividad completada (Gris).- Actividad en desarrollo (Verde).- Actividad no completada (Rojo).



5.1.2. Requerimientos no funcionales.

Tabla 2, Requerimientos no funcionales.

Número	Requisito No Funcional	Descripción
RFN-1	Disponibilidad	El sistema tiene que tener una disponibilidad constante de los recursos, para que el cliente pueda consultar en cualquier momento del estado de su proyecto.
RFN-2	Seguridad	La vista web debe proteger las operaciones que se realizan dentro de ella, para que estas se mantengan entre el usuario y el servidor, a través de la utilización de certificados de seguridad que aseguran el intercambio seguro entre ellos.
RFN-3	Extensible	La vista web tiene que tener la capacidad de implementar nuevos módulos de manera simple y sin complicaciones.
RFN-4	Tecnologías implementadas.	El sistema hará uso de angular y Node.js, los cuales serán desarrollados mediante los lenguajes de programación de JavaScript y Typescript.
RFN-5	Compatibilidad	La aplicación web tiene que tener la capacidad de visualizarse correctamente y realizar sus operaciones correspondientes en SmartPhones y en PCs (Independiente del sistema operativo).
RFN-6	Autenticación	Para visualizar los elementos correspondientes, el servidor necesitará confirmar las credenciales del usuario consultado previamente.

5.1.3. Acta de acuerdo.

Yo **BERNARDO CHOQUE TOLMO** en representación de **KUVEMAR SPA-**, en adelante cliente usuario del proyecto "**Desarrollo e integración de módulo para la presentación de estados de proyectos a clientes**". Estoy de acuerdo con los requisitos planteados en este documento y autorizo al equipo de software compuesto por **José Díaz** y **Cristian Fritis** ambos estudiantes de la carrera **Ing. Civil en computación e informática** de la **Universidad de Tarapacá**, el desarrollo del sistema (subsistema o aplicación) sugerido.

Firma del Cliente



5.2. Carta Gantt.

A continuación se presenta la carta gantt, la cual será utilizada para llevar a cabo el presente proyecto.

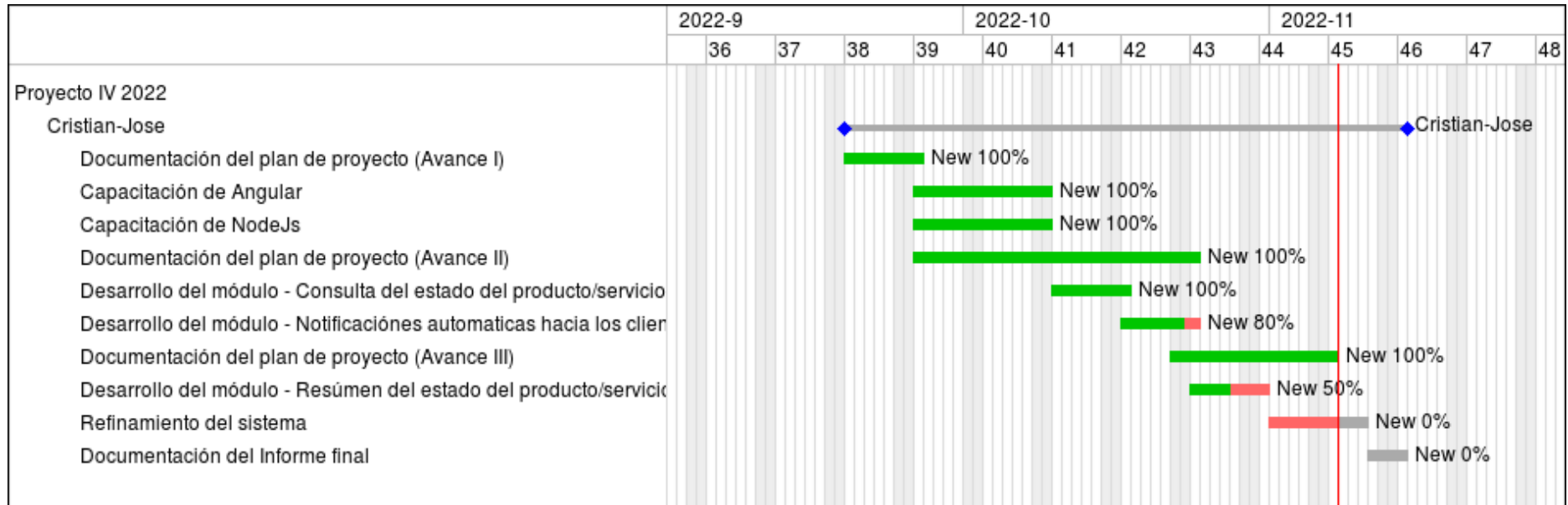


Figura 2, Carta Gantt del proyecto.

5.3. Modelo de contexto.

En el siguiente punto se puede apreciar el modelo de contexto, donde se muestra la interacción del sistema con todos sus subsistemas y el flujo de los datos que recorren entre todo el sistema, se tiene como autores del sistema al administrador que se va a encargar de tener todos los datos del sistema y el cliente que tiene la capacidad de ver los datos de su proyecto.

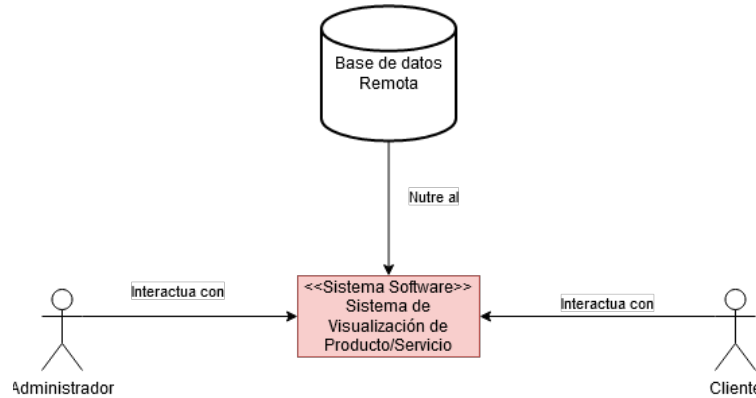


Figura 3, Diagrama de contexto.

5.3.1. Diagrama de comunicación subsistemas.

Al realizar un análisis del sistema, se identificaron los siguientes subsistemas.

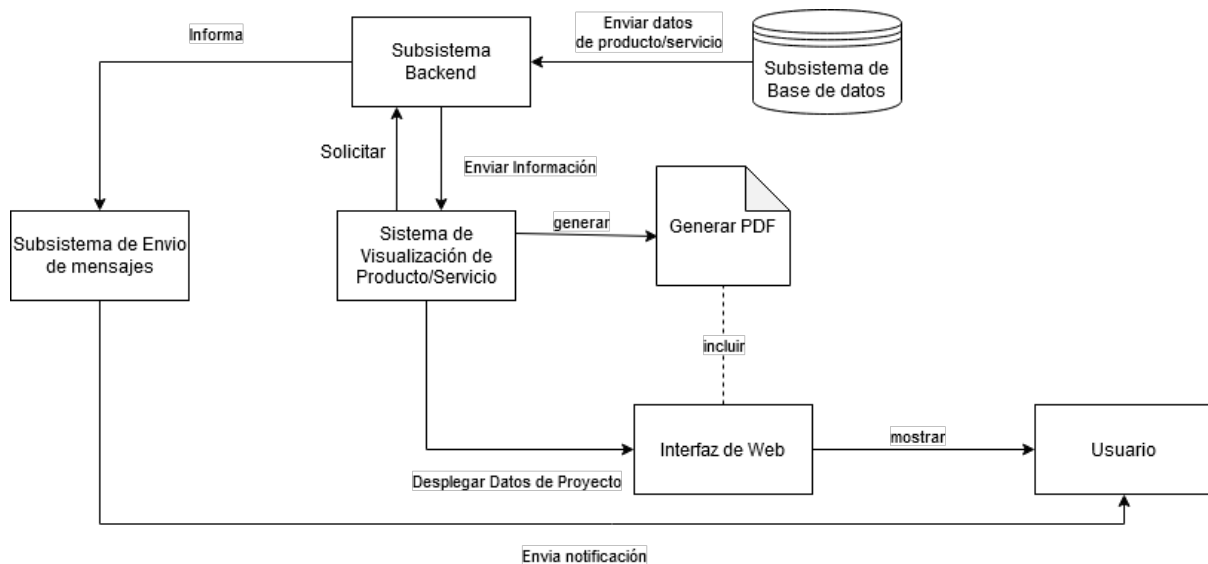


Figura 4, Diagrama de comunicación entre subsistemas.

Definición de los subsistemas:

- **Subsistema de Envío de Mensaje:** Encargado de realizar el envío de notificaciones por correo al cliente del proyecto de manera periódica.
- **Subsistema de Base de datos:** Se encarga de nutrir al sistema con la información del proyecto, la cual el cliente está consultado.
- **Interfaz Web:** Se encarga de visualizar todos los elementos cargados por el subsistema de base de datos.
- **Subsistema Backend:** Se encargará de toda la parte de la comunicación de los datos con el servidor base de datos y con el subsistema de notificación.

5.3.2. Diagrama de interacción entre subsistemas.

El siguiente diagrama representa la interacción existente entre los subsistemas que fueron declarados anteriormente en el punto 5.3.1.

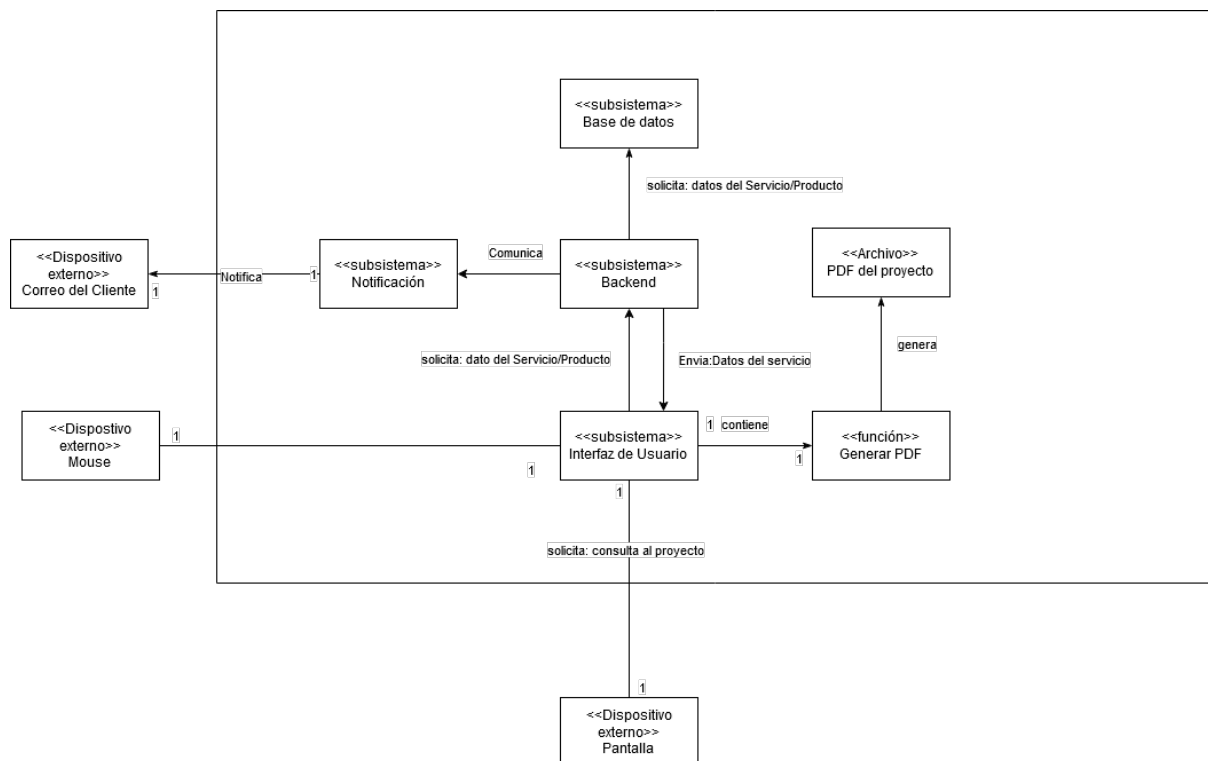


Figura 5, Diagrama de Interacción.



5.4. Gestión del proyecto.

5.4.1. Herramientas a utilizar.

Las herramientas que se van a utilizar para el desarrollo del proyecto llamado "**Desarrollo e integración de módulo para la presentación de estados de proyectos a clientes**".

- **Visual Studio Code:** Editor de código fuente que se utilizará para desarrollo de código de página web del proyecto.
- **GitHub:** herramienta de alojamiento para el control de versiones y colaboración de código fuente.
- **Trello:** Es una herramienta visual que permite a los equipos gestionar cualquier tipo de proyecto y flujo de trabajo, así como supervisar tareas o actividades del proyecto.
- **Angular:** Framework para el desarrollo de aplicaciones webs desarrollado con TypeScript.
- **NodeJS:** Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos.
- **CRON:** cron es un administrador regular de procesos en segundo plano que ejecuta procesos o funciones a intervalos regulares.
- **Uizard:** Es una herramienta para el diseño de interfaz de usuario. La principal característica es que permite la colaboración con otros usuarios.
- **Discord:** Herramienta de comunicación con el equipo de trabajo, aquí se realizarán las reuniones informales con los integrantes del equipo.
- **Zoom:** es una plataforma que ofrece servicios de videollamada a través de internet, está disponible para PC y Smartphone. Se utilizará para la comunicación con el cliente.
- **Redmine:** Lugar de almacenamiento remoto del proyecto, aquí estarán todos los entregables.
- **MySQL:** sistema de gestión de base de datos que se utilizara para el proyecto.



5.4.2. Framework Angular.

Angular es un Framework de JavaScript de código abierto escrito en **TypeScript**. Su objetivo principal es desarrollar aplicaciones de una sola página. Google se encarga del mantenimiento y constantes actualizaciones de mejoras para este framework.

Características de Angular:

- **TypeScript:** El framework utiliza TypeScript, la cual es un superset de Javascript fuertemente tipado, la cual esto hace que los otros usuarios que estén encargados en el desarrollo del código puedan entender el funcionamiento y el tipo de datos que utiliza o transporta las variables definidas. Todo el código de TypeScript después es compilado a formato Javascript para el despliegue en formato Web.
- **DOM (Document Object Model):** Se refiere a un documento XML o HTML, que tiene una estructura de árbol en la que cada nodo representa una parte del documento. Angular hace uso de DOM regular, lo que permite una mejor organización conforme avanza el desarrollo web.
- **Data binding (enlace de datos):** Es el proceso el cual se encarga de manipular elementos de una página web a través de un navegador.

Entre otras de las características de Angular, tenemos que su arquitectura se basa en bloques de construcción que son llamados Ng Modules, estos mismos proporcionan un contexto de compilación para los componentes que se vayan implementando durante el proyecto mismo, estos mismos, pueden ser reutilizados en otros proyectos a por venir. Lo mismo se puede apreciar de manera exacta al momento de verificar como es la arquitectura de un proyecto de Angular, donde tenemos un componente raíz que permite el arranque del mismo proyecto.

Lo anterior se puede apreciar en la figura N, donde tenemos la estructura de las carpetas generadas para realizar un proyecto en Angular. Inicialmente, como anteriormente se mencionó, existe el componente principal llamado app, en donde se tienen importados los módulos y componentes con los cuales funcionará la aplicación.

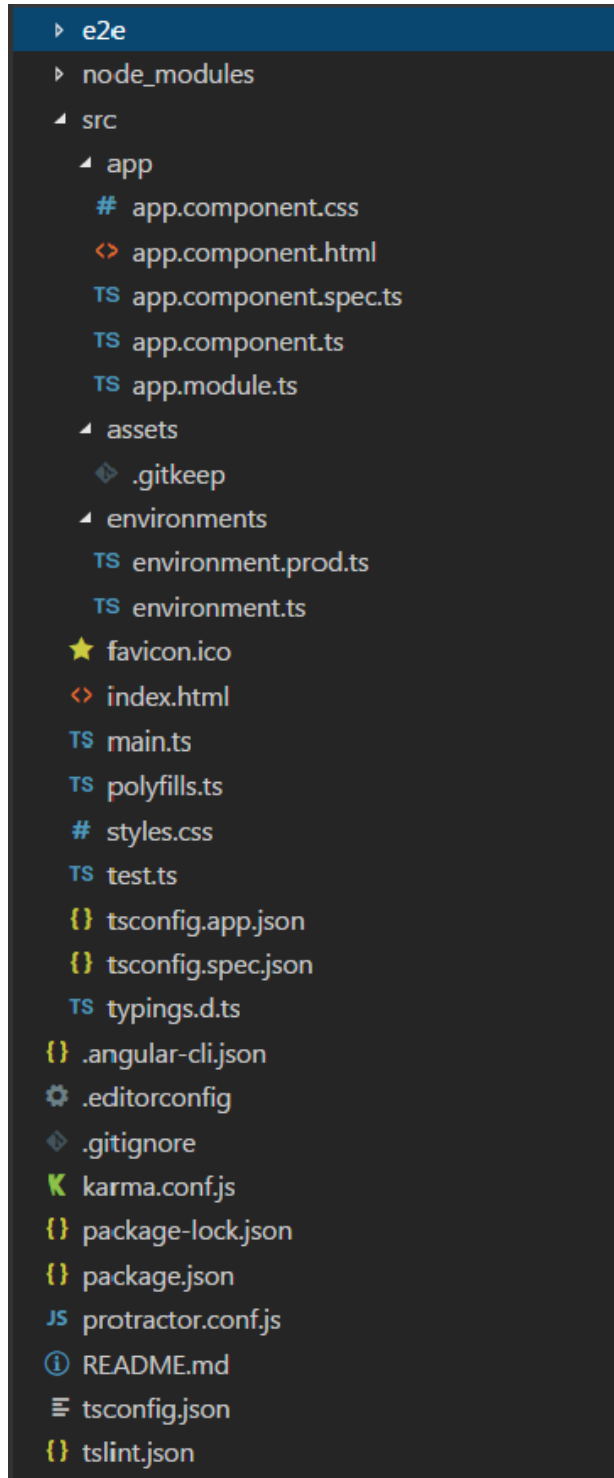


Figura 6, Estructura de carpetas Angular.



5.4.3. Framework NodeJS.

Nodejs es un *framework* para implementar operaciones de entrada y salida. Está basado en eventos, *streams* y construido encima del motor de Javascript V8, que es con el que funciona el Javascript de Google Chrome. En NodeJS se pueden desarrollar aplicaciones en JavaScript, la cual esencialmente se ejecuta en un equipo servidor.

Características de NodeJS:

- Generar páginas dinámicas en un servidor web.
- Acceder a bases de datos relacionales (MySQL, SqlServer etc.) y no relacionales (MongoDB, CouchDB, Cassandra etc.)
- Crear, leer y escribir archivos.
- Recuperar datos de formularios HTML.
- Procesar y almacenar archivos enviados desde una página web.
- Adaptable: Generar y enviar a aplicaciones cliente (navegador web, aplicación móvil etc.) archivos JSON
- Interactuar con sitios de una sola página (SPA) creados en Angular, React, Vue etc.
- Realizar envío de peticiones o respuesta a cualquier aplicación que realice alguna solicitud o envío de dato.
- Asíncronica: Permite el manejo de múltiples funciones a tiempo real sin que estén afecten el resultado o producto de cada una.
- Tiene la capacidad de comunicarse a tiempo real con el usuario o cliente, la cual esté solicitando de sus servicios.



Figura 7, Logo NodeJs.

5.5. Modelamiento de la aplicación.

5.5.1. Identificación del producto/servicio.

En esta pantalla el cliente puede consultar el estado de su producto/servicio, ingresando un código especial entregado por el administrador, en caso de que el usuario haya eliminado o perdido por accidente este código se puede comunicar con la empresa para solicitar el reenvío del código.

**¿Deseas ver como va el estado de tu
Producto/servicio contratado?**

Producto/Servicio a identificar

Buscar

**¿No encuentras tu pedido?
¡Escribenos!**





Dirección
Membrillar #2623. Arica –
Chile



Correo electrónico
info@kuvemar.com



Teléfono
(+56) 9 78073969

Figura 8, Identificación Producto/servicio.

5.5.2. Despliegue del estado de un producto/servicio.

A la hora de ingresar el código el usuario es redireccionado a esta pantalla, donde todos los datos del producto/servicio son desplegados, tales como el estado general de las actividades realizadas, el nombre del producto/servicio, poder generar .pdf con todos los datos generados, etc.

Por el momento se mostrará el Dashboard, que se encuentra en desarrollo.

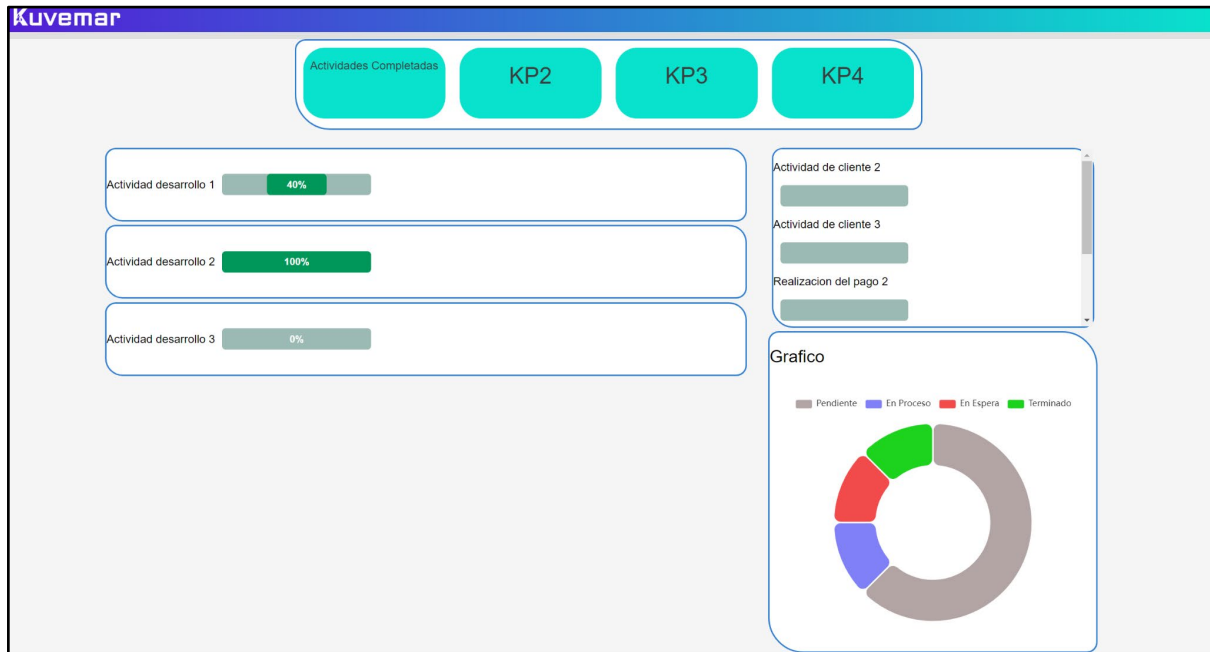


Figura 9, Dashboard del estado de un producto/servicio.



5.6. Herramientas a utilizar.

Las herramientas que se van a utilizar para el desarrollo del proyecto llamado "**Desarrollo e integración de módulo para la presentación de estados de proyectos a clientes**".

- **Visual Studio Code:** Editor de código fuente que se utilizara para desarrollo de código de página web del proyecto.
- **GitHub:** herramienta de alojamiento para el control de versiones y colaboración de código fuente.
- **Trello:** Es una herramienta visual que permite a los equipos gestionar cualquier tipo de proyecto y flujo de trabajo, así como supervisar tareas o actividades del proyecto.
- **Angular:** Framework para el desarrollo de aplicaciones webs desarrollado con TypeScript.
- **NodeJS:** Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos.
- **CRON:** cron es un administrador regular de procesos en segundo plano que ejecuta procesos o funciones a intervalos regulares.
- **Uizard:** Es una herramienta para el diseño de interfaz de usuario. La principal característica es que permite la colaboración con otros usuarios.
- **Discord:** Herramienta de comunicación con el equipo de trabajo, aquí se realizarán las reuniones informales con los integrantes del equipo.
- **Zoom:** es una plataforma que ofrece servicios de videollamada a través de internet, está disponible para PC y Smartphone. Se utilizará para la comunicación con el cliente.
- **Redmine:** Lugar de almacenamiento remoto del proyecto, aquí estarán todos los entregables.
- **MySQL:** sistema de gestión de base de datos que se utilizara para el proyecto.

5.7. Arquitectura del sistema.

A continuación, se presenta la arquitectura del sistema que presentara el proyecto, teniendo en cuenta las relaciones existentes entre la capa de datos, aplicaciones y presentación hacia el usuario.

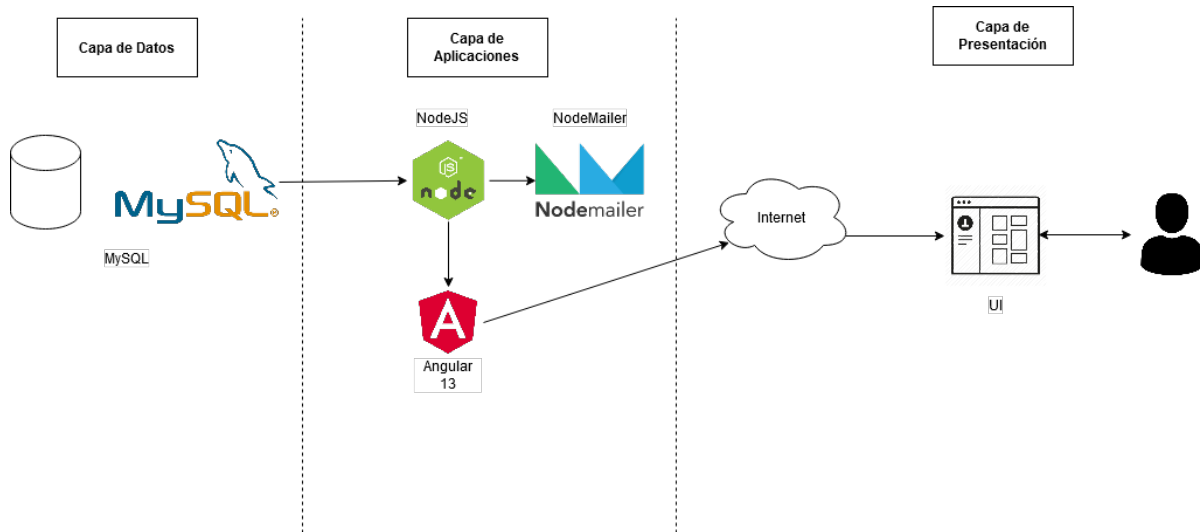


Figura 10, Arquitectura del software a utilizar para el proyecto.

5.8. Diagramas BPMN.

5.8.1. Despliegue estado de un producto/servicio.

En el siguiente diagrama BPMN, se busca conceptualizar el flujo que seguirá el sistema, para desplegar el estado de un producto/servicio hacia un cliente.

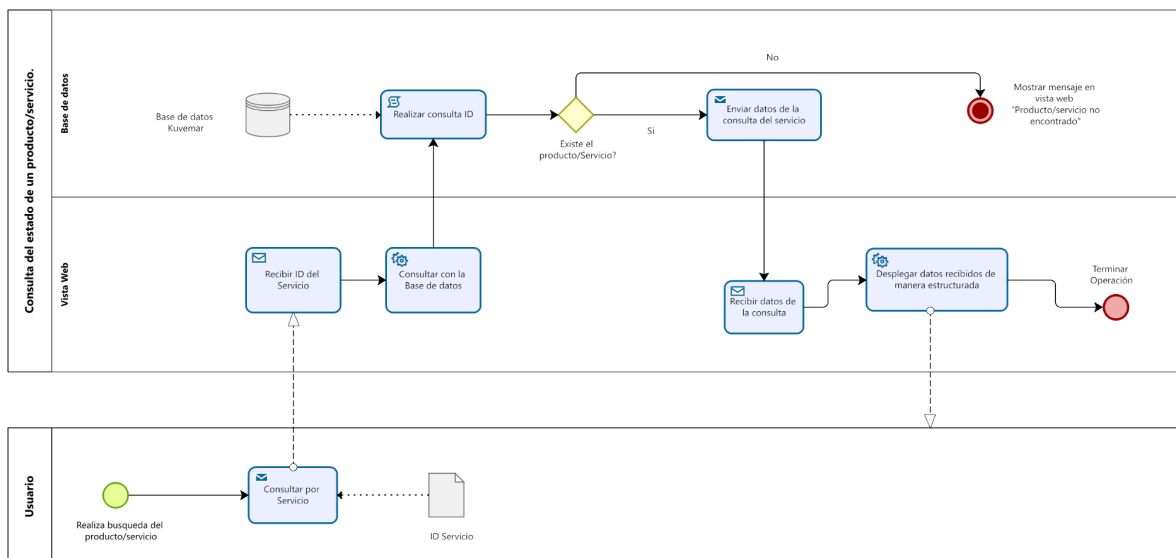


Figura 11, Diagrama BPMN "Despliegue estado de un producto/servicio".

5.8.2. Generar PDF.

A continuación, el siguiente diagrama representará el flujo de comunicación que sucederá al momento de que el cliente, busque crear un resumen de las actividades de su proyecto en un archivo .pdf.

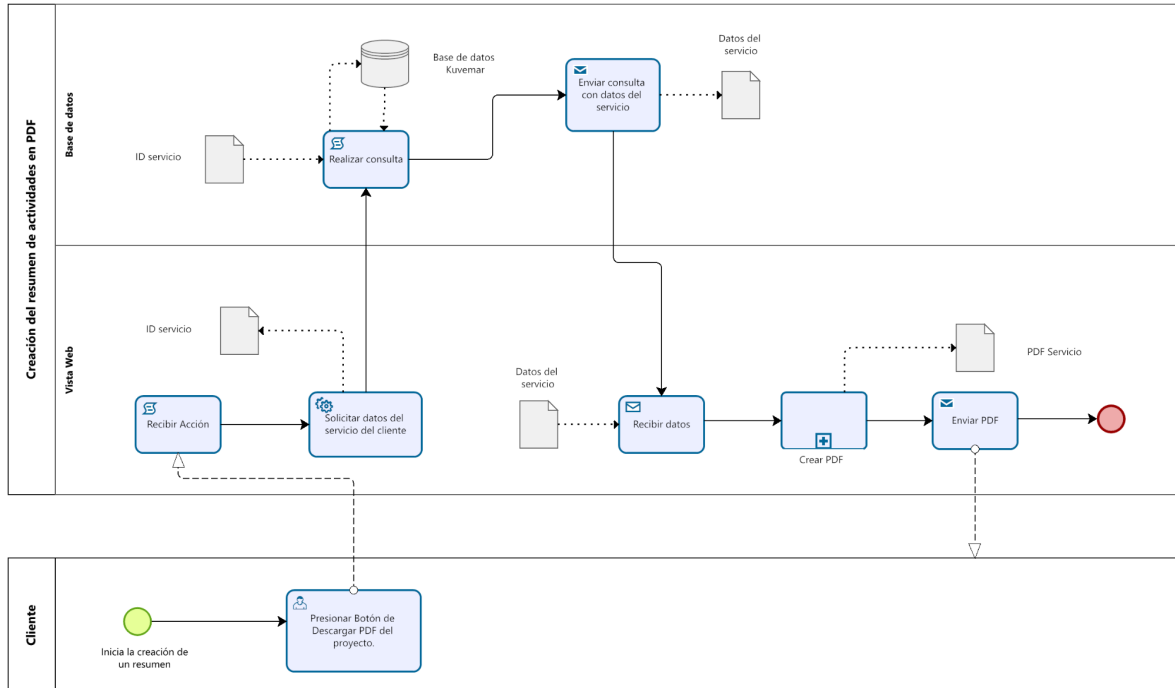


Figura 12, Diagrama BPMN "Generar PDF".

5.8.3. Notificación automática.

El siguiente diagrama busca modelar el manejo de las notificaciones automáticas por parte del Backend de nuestro sistema.

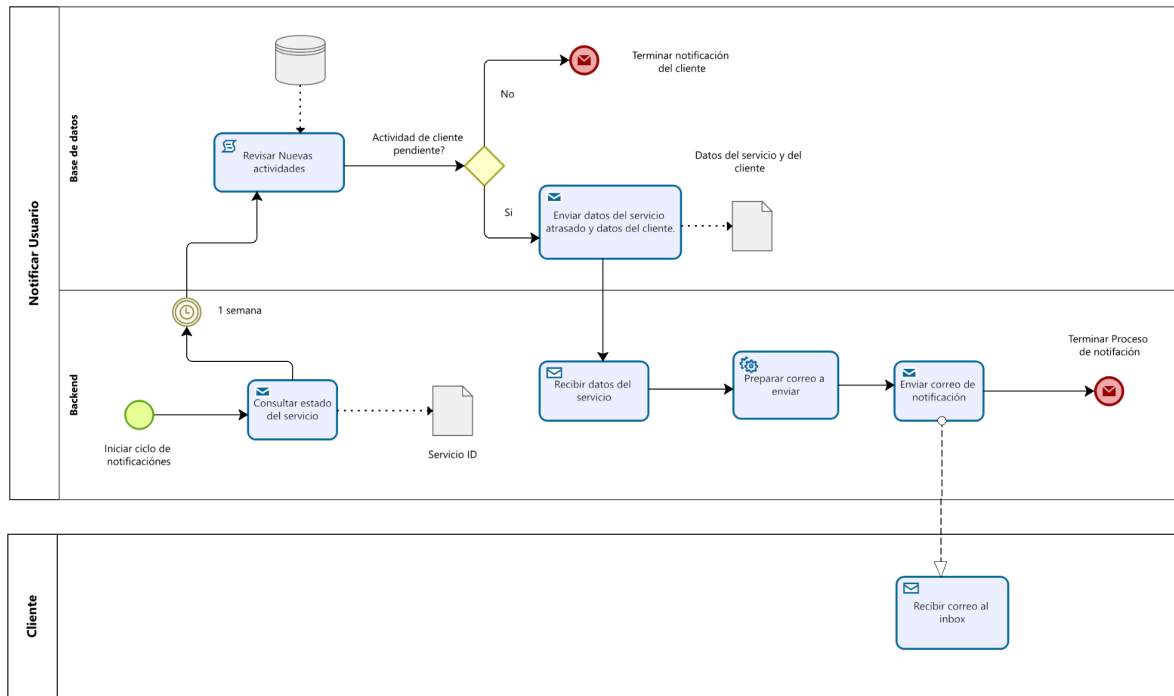


Figura 13, Diagrama BPMN "Notificación a usuarios."

5.9. Casos de Usos de Sistema

En esta sección se estarán describiendo todas las funcionalidades que tendrá el proyecto, la cual necesitará cumplir para el cliente.

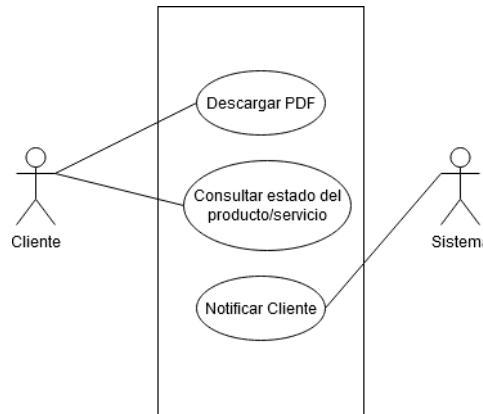


Figura 14, Casos de uso del sistema.

5.9.1. Descripción casos de Usos de Sistema

Tabla 3, Caso de uso "Descargar PDF".

Nombre	Descargar PDF
Descripción	Generar un archivo .pdf con el resumen del servicio/producto que el cliente está solicitando.
Precondición	Estar en una consulta de estado de un producto/servicio.
Actor	Cliente (principal), Sistema
Flujo Normal	
<i>Cliente</i>	<i>Sistema</i>
1. Apretar el botón de Descarga PDF que se encuentra en la interfaz web. 5. Descargar/Abrir Archivo PDF en la computadora.	2. Enviar solicitud de Generar PDF. 3. Crear de forma estructura el resumen del PDF, la cual contendrá todos los datos del servicio que se solicita. 4. Enviar Archivo PDF.
Postcondición	Archivo generado con éxito



Tabla 4, Caso de uso "Consultar estado del producto/servicio".

Nombre	Consultar estado del producto/servicio
Descripción	Mostrar al cliente el estado de su producto/servicio, que es cargado mediante una base de datos.
Precondición	Tener el ID del servicio/producto con anterioridad.
Actor	Cliente (principal), Sistema
Flujo Normal	
Cliente	Sistema
1. Ingresar el ID del producto/servicio correspondiente al usuario.	2. Capturar ID ingresando por parte del usuario. 3. Crear una consulta buscando el producto/servicio en la tabla de Base de datos MySQL, junto a todas sus actividades correspondientes. 4. Ejecutar consulta 5. Mandar respuesta al lado cliente de la aplicación. 6. Mostrarlo de manera ordenada en la interfaz Web del usuario.
Flujo Alternativo	
<i>Cliente</i>	<i>Sistema</i>
	3.1. En caso de que el ID del servicio/producto es incorrecta o no existe en la base de datos, realizar. 4.1. En caso de que no se tenga conexión / la base de datos no exista, mostrar error en consola. 6.1. En caso de que el ID recibido es incorrecto o ha ocurrido algún error en la transición de la petición, entonces mostrar en la interfaz que no se ha podido mostrar el contenido.
Postcondición	Despliegue del producto/servicio con éxito.



Tabla 5, Caso de uso "Notificar cliente".

Nombre	Notificar Cliente
Descripción	Notificar al cliente de las actividades de cliente que falta por resolver.
Precondición	Tener una actividad de cliente pendiente.
Actor	Cliente, Sistema(principal)
Flujo Normal	
Cliente	Sistema
	<ol style="list-style-type: none">1. Con un programa planificador de tareas, semanalmente comprobar que no tenga actividad de cliente pendiente.2. En caso de que exista una actividad pendiente, redactar un mensaje por correo.3. Realizar consulta en la base de datos del proyecto en cuestión con el ID guardada del producto/servicio local.4. Realizar consulta del estado del producto/servicio.5. Enviar consulta realizada.6. Transformar la consulta y agregar la información conseguida al cuerpo de la carta del cliente.7. Enviar carta al cliente.
8. Carta recibida con éxito.	
Flujo Alternativo	
Cliente	Sistema
	<ol style="list-style-type: none">3.1. En caso de que no exista, no enviar ningún mensaje al cliente.1.1 y 3.2. Si es que no existe conexión a la base de datos, dejar en espera hasta que se pueda conectar.7.1. En caso de que no se tenga conexión con el proveedor de correo, entonces mostrar error en consola.
Postcondición	El cliente recibe el correo con exitoso.

5.10. Diagramas de Secuencia.

5.10.1. Diagrama de secuencia “Despliegue producto/servicio”.

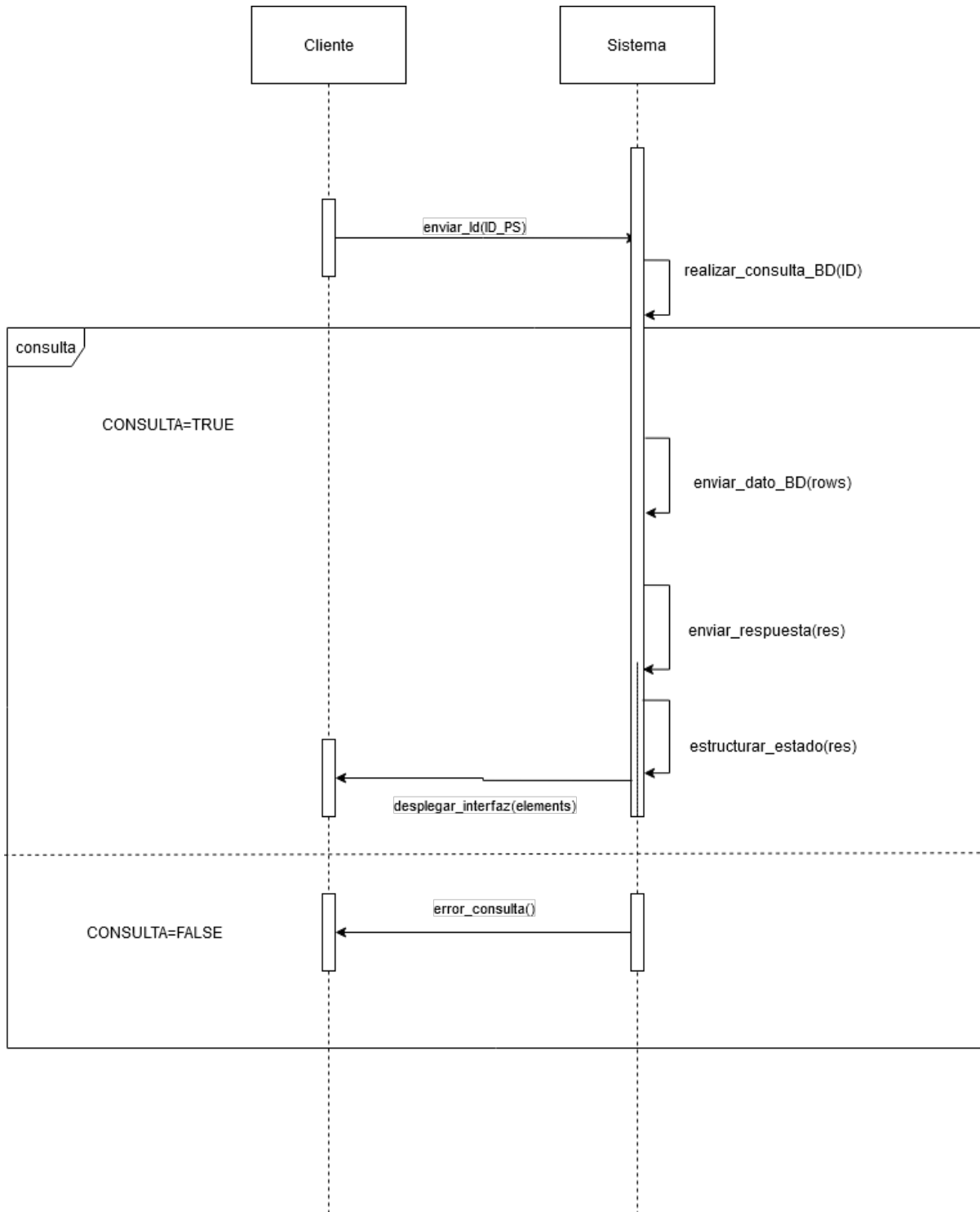


Figura 15, Diagrama de secuencia Nivel 0 - Despliegue de estado de producto/servicio.

5.10.2. Diagrama de secuencia “Notificaciones automáticas”.

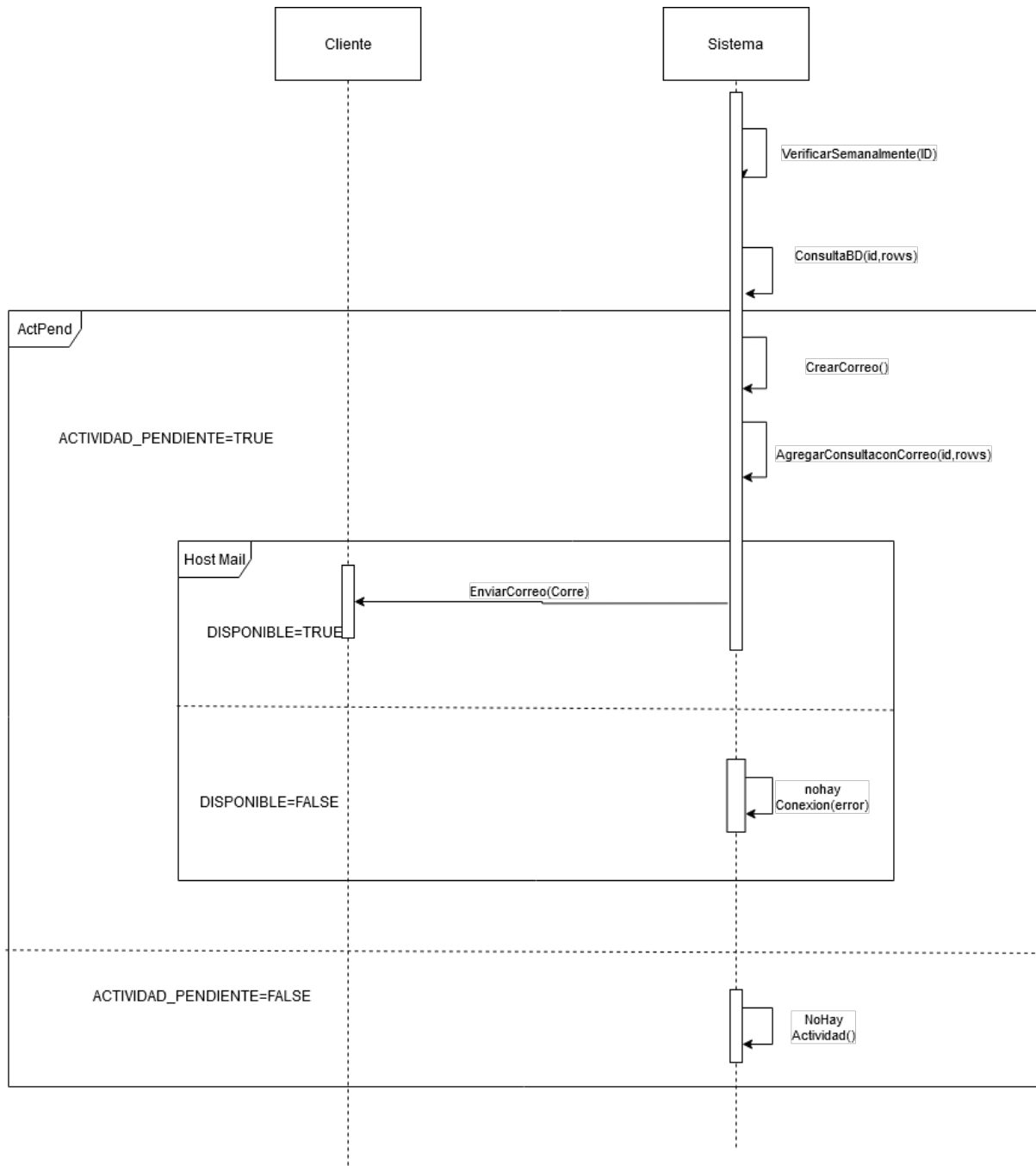


Figura 16, Diagrama de secuencia Nivel 0 - Notificación por Correo del estado del servicio.

5.10.3. Diagrama de secuencia “Creación de PDF”.

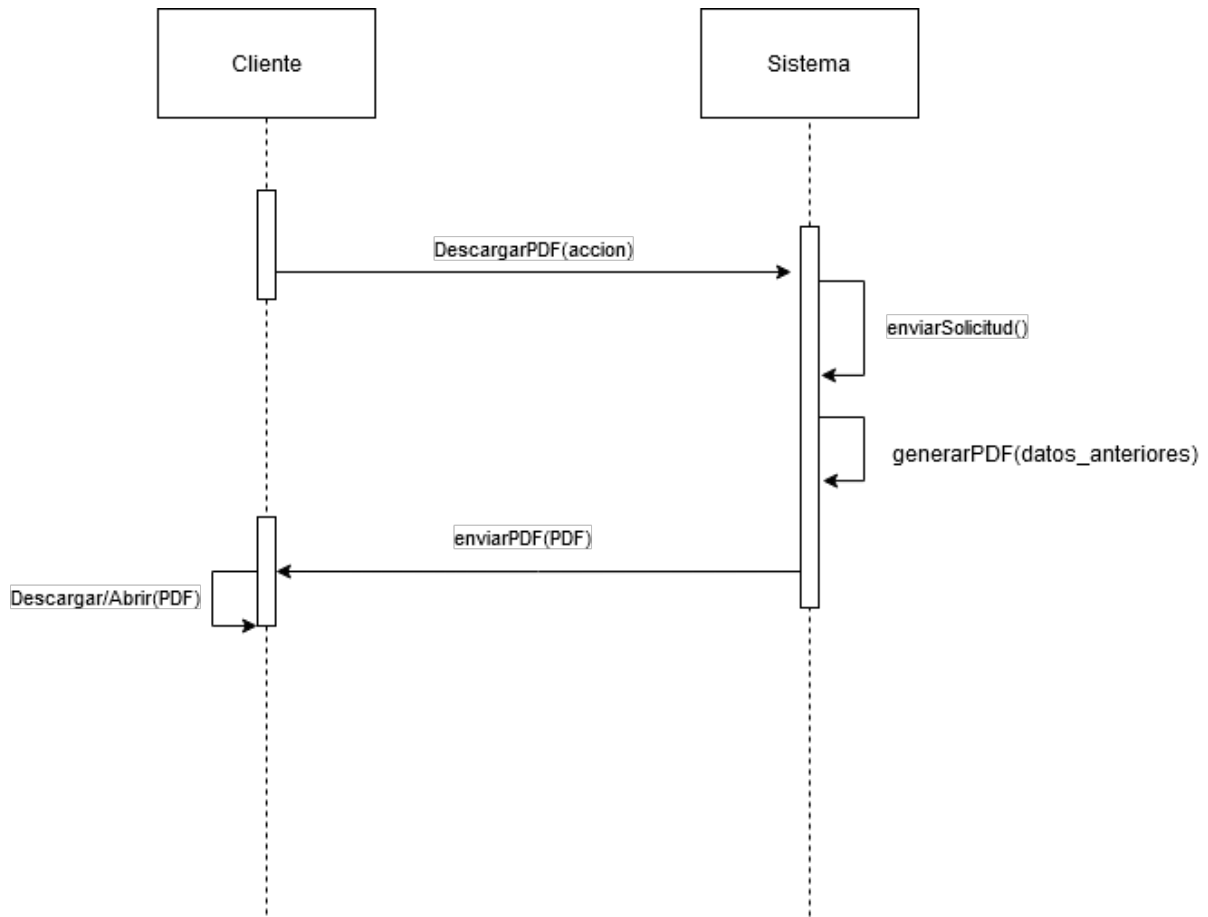


Figura 17, Diagrama de secuencia Nivel 0 - Descargar PDF.



5.11. Base de datos.

Para el apartado de la base de datos, solo se ocupó una pequeña porción de la que originalmente es ocupada para gestionar los proyectos desde la empresa Kuvemar. Esta sección de la base de datos está conformada por 2 entidades que están unidas por una relación 1 a N, donde se tiene N actividades por 1 producto/servicio.

- Entidad “servicios”:

Entidad que representa a los servicios/productos que se están desarrollando, esta entidad contiene los siguientes atributos; id(Pk), nombre, descripción, valorTotal, moneda, fechaInicio, fechaTermino, fechaVencimiento, tipo_servicio, usuarios_id_cliente, usuarios_id_encargado, estado.

- Entidad “actividades”:

Entidad que representa a las actividades que contiene cada servicio, esta entidad contiene los siguientes atributos; id(PK), nombre, estado, fechaVencimiento, fechaInicio, fechaTermino, descripción, relevancia, importante, comentarioFinal, usuarios_id, servicios_id (FK de Servicios), tipo_servicios_id, plantilla, tipo_actividad, progreso, prioridad, dificultad, posición.

El siguiente diagrama Entidad Relación, vendría a representar las entidades descritas.

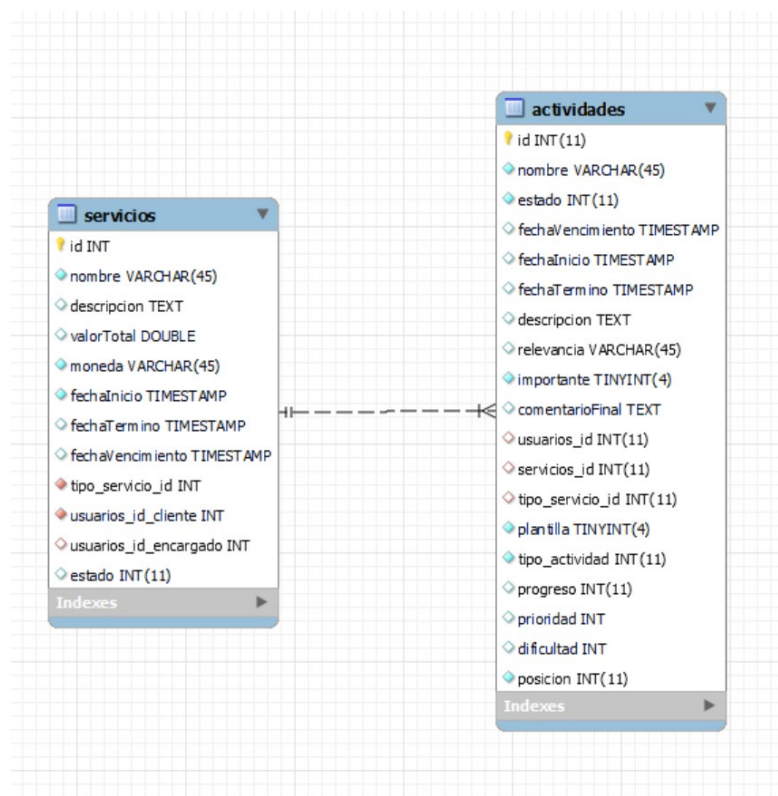


Figura 18, Sección base de datos Kuvemar.



5.12. Repositorio

Para el concepto de gestión más eficiente, colaborativo y controlado del proyecto, se ha decidido utilizar un repositorio remoto, que en este caso sería GitHub (Aplicación github desktop). La cual nos otorga una serie de funciones de subida y bajada de todo el código que se va implementando con el equipo de desarrollo, como también realizar un control de las versiones en caso de que se produzca algún fallo en el transcurso del proyecto.

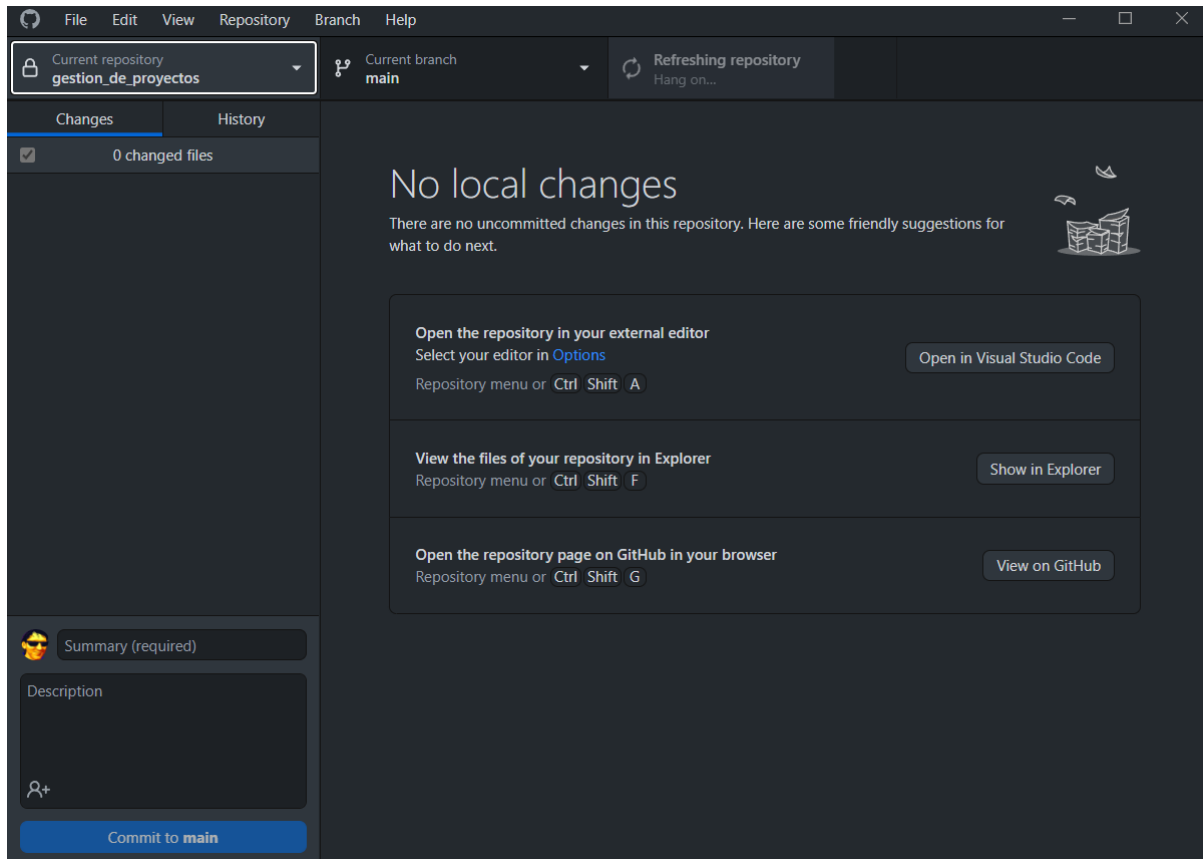


Figura 19, GitHub desktop del Proyecto.



6. Conclusiones.

En el informe correspondiente se dio a definir el principal problema y su correspondiente solución para la realización del proyecto, dando a definir los correspondientes requerimientos funcionales y no funcionales, dando a definir una carta Gantt con la cual se trabajará para realizar el presente proyecto.

También se ha logrado tener una buena comunicación con el cliente, llegando a establecer un horario establecido para realizar reuniones semanales, para ir informando sobre los avances que se van realizando en el proyecto.

El funcionamiento de cada módulo del proyecto se ha definido de manera correcta, para desarrollar los mismos, explicando los diferentes casos que pueden ocurrir en cada pieza que componen los mismos, las diferentes funciones que tienen cada uno y los diferentes casos alternativos a la hora de que sucede algún imprevisto que no sea el caso normal.

Otro punto importante que se ha podido realizar, son las diferentes interacciones de los subsistemas que compondrán el proyecto, logrando una buena comunicación entre todos los modelos a implementar.

Como punto relevante, se ha visto que la metodología scrum es muy valioso para el avance de estos tipos proyectos, ya que utilizando esa metodología se podrá mantener informado al cliente de los distintos avances en las actividades del proyecto y del producto en general.

Por último, se ha decidido implementar el código fuente del proyecto a un repositorio para tener más controlado los cambios que se van realizando durante el transcurso del proyecto, como también para realizar copias de seguridad en caso de que produzcan algún fallo a la hora de desarrollar el producto solicitado.