

Universidad Siglo 21



Trabajo Final de Grado. Prototipado Tecnológico

Carrera: Licenciatura en informática

Sistema de seguimiento de proyectos e incidencias

Autor: Dib Juan Matías

Legajo: VINF05864

Córdoba, Noviembre de 2021

## Índice

<b>Título</b>	1
<b>Introducción</b>	1
<i>Descripción del área problemática</i>	2
<b>Justificación</b>	2
<b>Objetivo general</b>	3
<b>Objetivos específicos</b>	3
<b>Marco teórico referencial</b>	3
<i>Dominio del problema</i>	3
<i>Tic</i>	4
<i>Competencias</i>	6
<b>Diseño metodológico</b>	8
<i>Herramientas metodológicas</i>	8
<i>Herramientas de desarrollo</i>	9
<i>Recolección de datos</i>	10
<i>Planificación del proyecto (Gantt)</i>	10
<b>Relevamiento</b>	12
<i>Relevamiento estructural</i>	12
<i>Relevamiento funcional</i>	12
<i>Organigrama</i>	12
<b>Procesos de negocios</b>	15
Objetivo, límites y alcance del Prototipo.	18
Objetivo del prototipo.	18
Límites del prototipo.	18
Alcance del prototipo.	18

Descripción del sistema.	19
Requerimientos no funcionales.	20
Diagrama de casos de usos	21
Descripción de casos de uso	26
Diagrama de secuencia	37
Estructura de datos	39
Diagrama entidad relación (DER)	40
Prototipo de interfaz de pantalla	40
Diagrama de despliegue	43
Bibliografía	43

## **Resumen**

El presente trabajo final de carrera está enfocado a la necesidad por la que pasan los equipos de trabajo en la gestión de proyectos de desarrollo de software, al momento de planificar y administrar sus actividades. Como solución a estos problemas, aparecen herramientas destinadas a la autogestión y organización, es decir, que las mismas ayudan a establecer un orden para aquellas actividades y tareas a realizar, para establecer el orden de aquellas tareas por realizar, las que están en progreso y asimismo las que van finalizando, dejando por sentado cada movimiento dentro del proyecto y manteniendo al tanto de cada situación a todos los miembros que pertenezcan a él, teniendo como prioridad la trazabilidad del proyecto.

Por lo tanto, en base a una investigación exhaustiva y a la problemática planteada, se presenta una propuesta de herramienta basada en una plataforma digital, destinada a mejorar y optimizar el flujo de trabajo, permitiendo la gestión flexible del mismo y aportando funciones específicas para entregar valor agregado a clientes, de forma tal que no sólo sea utilizada para equipos de desarrollo.

Palabras clave: desarrollo de software, autogestión de tareas, plataforma digital, equipos de trabajo, trazabilidad del proyecto.

## Abstract

This final degree project is focused on the need that work teams go through in managing software development projects, when planning and managing their activities. As a solution to these problems, tools for self-management and organization appear, that is, they help to establish an order for those activities and tasks to be carried out, to establish the order of those tasks to be carried out, those that are in progress and also those that are finalizing, taking for granted each movement within the project and keeping all the members who belong to it aware of each situation, having as priority the traceability of the project.

Therefore, based on an exhaustive investigation and the problem raised, a tool proposal based on a digital platform is presented, aimed at improving and optimizing work flow, allowing flexible management of it and providing specific functions to deliver value. added to clients, in such a way that it is not only used for development teams.

**Keywords:** software development, self-management of tasks, digital platform, work teams, project traceability.

## **Título**

Sistema de seguimiento de proyectos e incidencias

## **Introducción**

El mundo de los negocios y las empresas ha cambiado. Lo ha hecho toda la sociedad en general, pero en el sector empresarial, ahora los trabajadores y CEO's de todo el mundo se enfrentan a nuevos panoramas laborales caracterizados por el teletrabajo, la distribución de tareas y la sinergia entre departamentos.

El presente proyecto trata de una plataforma web orientada a equipos de todo tipo, para la administración de tareas de un proyecto y flujos de trabajo, seguimiento de errores e incidencias.

### *Antecedentes*

El desarrollo de los sistemas tradicionales de ciclo de vida se originó en la década de 1960 para desarrollar a gran escala funcional de sistemas de negocio en una época de grandes conglomerados empresariales.

La idea principal era continuar el desarrollo de los sistemas de información en una muy deliberada, estructurada y metódica, reiterando cada una de las etapas del ciclo de vida. (Ingeniería de Software, 2011)

La evolución de la disciplina de ingeniería del software ha traído consigo propuestas diferentes para mejorar los resultados del proceso de construcción. Las metodologías tradicionales, pone especial énfasis en los procesos lineales, la documentación, la planificación por adelantado y la priorización. Con el método tradicional, el tiempo y el presupuesto son variables y los requerimientos (aquello que

tenemos que lograr) son fijos, debido a esto a menudo existen problemas de presupuesto y plazos, y las metodologías ágiles haciendo énfasis en la adaptabilidad del proceso, en el trabajo en equipo, la colaboración, las tareas y la flexibilidad para responder al cambio lo más rápido posible. (Herrera, 2009)

### *Descripción del área problemática*

Actualmente las empresas e instituciones están cada vez más interesadas en mantenerse actualizadas con las nuevas tecnologías para garantizar mayor productividad y eficiencia en la realización de sus proyectos. Esto puede resultar en un incremento constante de nuevos proyectos a los departamentos de TI de las empresas, sobrecargando al equipo de colaboradores y haciendo que tanto el desarrollo como el ambiente de trabajo se vuelva un caos.

Las metodologías tradicionales imponen una disciplina rigurosa de trabajo sobre el proceso de desarrollo de software. La definición del alcance no llega a cumplir con lo solicitado por el usuario y tampoco se llega con el tiempo establecido, lo cual trae como consecuencia un impacto negativo sobre el costo del desarrollo. En cuanto al aplicativo, existen incongruencias y el equipo de desarrollo no trabaja como debe de ser, ya que se presentan situaciones como el realizar modificaciones que no son informadas a todos los interesados y esto desemboca una falla de comunicación entre pares.

## **Justificación**

Con el sistema de información propuesto se pretende ayudar a equipos de todo tipo a gestionar el trabajo, optimizar la productividad lo cual trae impactos positivos a la organización ya que esto último genera un aumento en la rentabilidad del negocio, también adaptándose a cualquier tipo de proyecto, desde la simple gestión de tareas hasta iniciativas más complejas, promoviendo la reducción de errores, la mejora de comunicación en el equipo y aumentando la calidad para los clientes.

## Objetivo General Del Proyecto

Proponer un modelo de sistema web para gestión de proyectos que permita optimizar la productividad de los equipos de trabajo, llevar un seguimiento de errores, gestión de contenidos, planificación de tareas en un proyecto determinado.

## Objetivos Específicos

- Definir las funcionalidades necesarias para el sistema propuesto.
- Diseñar páginas y navegación de las mismas para el sistema propuesto.

## Marco teórico referencial

### *Dominio del problema*

Para comenzar a analizar el dominio del problema, se deben definir las cuatro fases: planeación, construcción, implementación y cierre.

Planeación: es la etapa en la que se diseña el proyecto, en esta fase las actividades deberán incluir hacer investigación para determinar las metas y los requisitos del proyecto.

Construcción: en esta fase se recluta el equipo y se les da las indicaciones sobre horarios, presupuesto, recursos del proyecto, etc. Por lo general se empieza con una reunión para explicar el plan del proyecto, analizar procesos y responder todas las preguntas que puedan surgir.

Implementación: Es en esta etapa donde el proyecto comienza a desarrollarse plenamente, donde es de suma importancia hacer un seguimiento de todas las tareas que se van llevando a cabo, presupuesto, emitir reportes de avance y manteniendo la comunicación con todo el equipo.

Cierre: Finaliza el proyecto. Aquí se evalúa el rendimiento Gral. y las aéreas que se pueden mejorar. (Una guía completa para la gestión de proyectos en 2021, s.f.)



*Tic*

Dentro del amplio mundo de las tecnologías de la información y comunicación, se desea presentar los siguientes conceptos para el desarrollo del presente proyecto:

Aplicaciones web: dentro de su arquitectura pueden distinguirse y mencionarse tres niveles, el nivel superior que interacciona con el usuario o cliente web, normalmente el navegador, el nivel inferior que proporciona los datos alojados en la base de datos, y el nivel intermedio que procesa los datos o también conocido como el servidor web. Para que estas aplicaciones web funcionen el cliente y servidor deben comunicarse utilizando un protocolo estandarizado conocido como HTTP, el cual forma parte de la familia de protocolos TCP/IP, que son empleados en internet. Estos protocolos permiten la conexión de sistemas heterogéneos, lo que facilita el intercambio de información entre distintos ordenadores. (Mora, 2002)

JAVASCRIPT: Es un lenguaje interpretado, basado en objetos (no es un lenguaje orientado a objetos puro) y multiplataforma, inventado por Netscape Communications Corporation.

JavaScript permite crear aplicaciones que se ejecuten a través de Internet, basadas en el paradigma cliente/servidor. La parte del cliente se ejecuta en un navegador, como Netscape Communicator o Microsoft Internet Explorer, mientras que la parte del servidor se ejecuta en un servidor, como Netscape Enterprise Server. (Mora, 2002)

HTML: Es un lenguaje de marcas (etiquetas) que se emplea para dar formato a los documentos que se quieren publicar en la WWW. Los navegadores son capaces de interpretar las etiquetas y mostrar los documentos con el formato deseado.

Las páginas web o páginas HTML son unos cheros escritos en el lenguaje HTML. El desarrollo de estas páginas abarca un amplio grupo de tecnologías, desde las páginas más sencillas que sólo usan el lenguaje HTML hasta las más complejas que usan DHTML, CSS, JavaScript, applets realizados en Java, etc.

El lenguaje HTML se basa en SGML, un sistema mucho más completo y complicado de procesamiento de documentos que indica como organizar y marcar (etiquetar) un documento. HTML define e interpreta las etiquetas de acuerdo a SGML. (Mora, 2002)

HTML5: No es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red.

Html5 provee básicamente tres características: estructura, estilo y funcionalidad. Es considerado el producto de la combinación de HTML, CSS y Javascript.

Propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones. (Gauchat, 2012)

NODE.JS: Ideado como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node.js está diseñado para crear aplicaciones network escalables.

Las redes basadas en hilos son relativamente ineficientes y muy difíciles de usar. Además, los usuarios de Node.js están libres de preocuparse por el bloqueo del proceso, ya que no existe. Casi ninguna función en Node.js realiza I/O directamente, por lo que el proceso nunca se bloquea. Por ello, es muy propicio desarrollar sistemas escalables en Node.js.

HTTP es un elemento destacado en Node.js, diseñado teniendo en cuenta la transmisión de operaciones con streaming y baja latencia. Esto hace que Node.js sea muy adecuado para la base de una librería o un framework web. (Node js, 2020)

REACT.JS: Ayuda a crear interfaces de usuario interactivas de forma sencilla. Diseña vistas simples para cada estado en la aplicación, y se encarga de actualizar y renderizar de manera eficiente los componentes correctos cuando los datos cambien las vistas declarativas hacen que tu código sea más predecible, por lo tanto, fácil de depurar.

Crea componentes encapsulados que manejen su propio estado, y los convierte en interfaces de usuario complejas, ya que la lógica de los componentes está escrita en JavaScript y no en plantillas, puedes pasar datos de forma sencilla a través de tu aplicación y mantener el estado fuera del DOM. (<https://es.reactjs.org/>, 2019)

**MYSQL:** Es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (Pecos, 2012)

### *Competencias*

**JIRA:** Es una aplicación basada en web para el seguimiento de errores, incidencias y la gestión operativa de proyectos, también se utiliza en áreas no técnicas para la administración de tareas. La herramienta fue desarrollada por la empresa australiana Atlassian. Inicialmente Jira se utilizó para el desarrollo de software, sirviendo de apoyo para la gestión de requisitos, seguimiento del estatus y más tarde para el seguimiento de errores. Jira puede ser utilizado para la gestión de procesos y para la mejora de procesos, gracias a sus funciones para la organización de flujos de trabajo. (Frias, 2002)

**ASANA:** Es una herramienta que permite gestionar el flujo de trabajo (tareas y proyectos) ofreciendo el detalle de inicio y culminación de cada una de estas, así mismo conecta a todo el equipo de trabajo permitiendo la comunicación, organización y planificación para llevar a feliz término los proyectos en cuestión.

Sirve para administrar y gestionar de forma centralizada y organizada todas las tareas y/o actividades a llevar a cabo en un proyecto, a través de ella es posible conformar equipos de trabajo, con los cuales se compartirá la planificación de cada una

de las tareas que deban llevar a feliz término para cumplir con los objetivos deseados. (Muradas, s.f.)

TRELLO: Es uno de los gestores de proyectos colaborativos más populares y más utilizados en Internet tanto para uso particular como de empresa. Está enfocado, sobre todo, a la coordinación con el resto de los miembros del equipo y permite administrar proyectos profesionales u organizar todo tipo de eventos, reuniones o tareas del día a día. Diseño metodológico.

Hay tareas, tarjetas y listas que podemos ir moviendo según prioridad o según se hayan ido cumpliendo. En las diferentes tarjetas podemos escribir y detallar las tareas y Trello también nos permite añadir fechas de vencimiento, conversaciones o planificaciones o archivos adjuntos que ayuden a controlar qué es cada una.

Una de las claves del éxito de Trello es que utiliza un sistema Kanban para la gestión de tareas que han heredado otras muchas aplicaciones y software similares. (García, 2021)

## **Diseño Metodológico**

### *Herramientas metodológicas*

Para llevar adelante el desarrollo de la aplicación, se ha optado por utilizar la metodología Kanban. Ya que es un método flexible y permite la división del trabajo en varias partes donde las actividades se visualizan en tarjetas plasmando el estado que se encuentra cada actividad sobre una pizarra. Esto permitirá un progreso ágil, visual y ordenado del proyecto.

KANBAN: Es un enfoque esbelto para el desarrollo de software ágil. En realidad, Kanban significa muchas cosas. Literalmente, Kanban es una palabra japonesa que significa "tarjeta visual". En Toyota, Kanban es el término utilizado para el sistema de señalización visual y física que une todo el sistema de producción just in time. La mayoría de los métodos ágiles como Scrum y XP ya están bien alineados con los principios LeanThinking. En 2004, sin embargo, David Anderson fue pionero en una

implementación más directa de Lean Thinking y Theory of Constraints para el desarrollo de software. Bajo la guía de expertos como Don Reinertsen, esto se convirtió en lo que David llamó un "sistema Kanban para el desarrollo de software", y al que la mayoría de la gente ahora simplemente se refiere como "Kanban". Entonces, si bien Kanban aplicado al desarrollo de software es nuevo, Kanban utilizado en la producción just in time tiene más de medio siglo. (Crisp (.))

Cómo funciona Kanban:

### 1. Visualiza el flujo de trabajo

- Divida el trabajo en partes, escriba cada elemento en una tarjeta y colóquelo en la pared.
- Utilice columnas con nombre para ilustrar dónde se encuentra cada elemento en el flujo de trabajo.

2. **Limitar WIP** (trabajo en progreso): asigne límites explícitos a la cantidad de elementos que pueden estar en progreso en cada estado del flujo de trabajo.

3. **Mida el tiempo de entrega** (tiempo promedio para completar un artículo, a veces llamado "tiempo de ciclo"), optimice el proceso para hacer que el tiempo de entrega sea lo más pequeño y predecible posible. (Crisp (.))

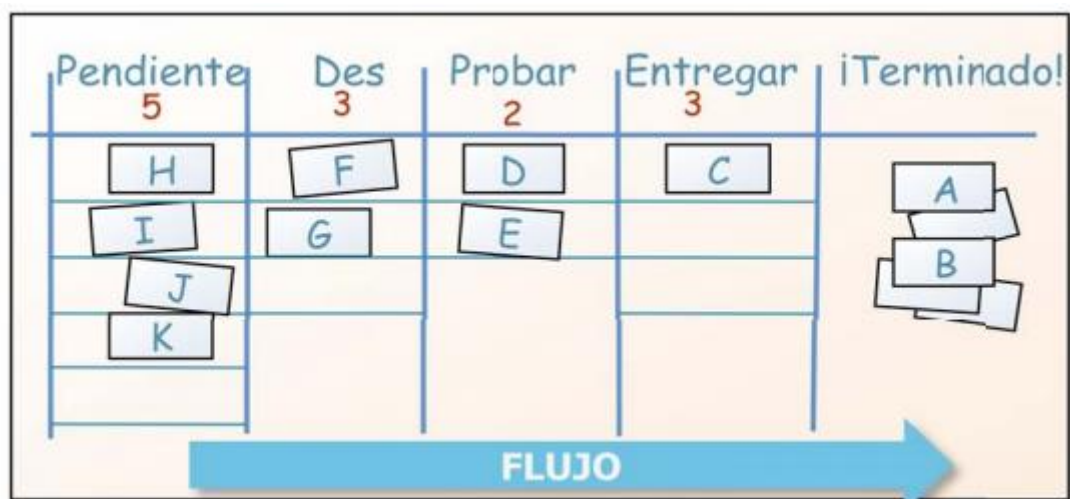


Figura tablero Kanban (Crisp, 2010)

### *Herramientas de desarrollo*

Para la parte del desarrollo front-end se opta por el marco de trabajo react.js debido a su capacidad para desarrollar interfaces mediante componentes donde cada componente contiene tanto la lógica como la parte visual, de este modo se puede reutilizar dentro de otros componentes combinándolo con código HTML y JavaScript.

Por otro lado, para back-end se utiliza una de las tecnologías más elegidas del momento que es la plataforma Node.js, que va a permitir aplicar los conocimientos de JavaScript.

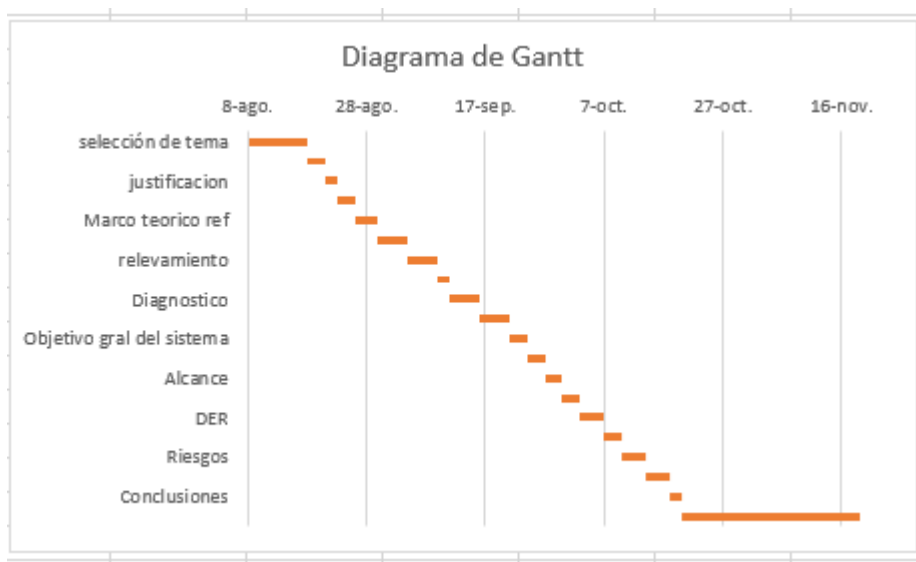
Y, por último, se utiliza como gestor de base de datos MySQL.

### *Recolección de datos*

Desde el punto de vista del proyecto, la técnica de observación ayudó a percibir el estado actual del mundo de los negocios y las empresas. De acuerdo con un estudio Pulse of the Profession realizado en 2017 por Project Manager Institute (PMI) El 71% de las organizaciones entrevistadas confirma que utiliza enfoques ágiles para sus proyectos, a veces, con frecuencia o siempre. En el último año se registró que 1 de cada 5 proyectos ha utilizado enfoques ágiles. Según el informe, las empresas ágiles incrementan los ingresos un 37 % más rápido, y generan un 30% más de ganancias frente a aquellas empresas que no se gestionan “ágilmente”. (Campos, 2018)

### *Planificación del proyecto (Gantt)*

selección de tema	8-ago.	10	18-ago.
introduccion	18-ago.	3	21-ago.
justificacion	21-ago.	2	23-ago.
objetivos	23-ago.	3	26-ago.
Marco teorico ref	26-ago.	4	30-ago.
Diseño metodologico	30-ago.	5	4-sep.
relevamiento	4-sep.	5	9-sep.
Proceso de Negocio	9-sep.	2	11-sep.
Diagnostico	11-sep.	5	16-sep.
Propuesta	16-sep.	5	21-sep.
Objetivo gral del sistema	21-sep.	3	24-sep.
Limite	24-sep.	3	27-sep.
Alcance	27-sep.	3	30-sep.
Descripcion del sistema	30-sep.	3	3-oct.
DER	3-oct.	4	7-oct.
Seguridad	7-oct.	3	10-oct.
Riesgos	10-oct.	4	14-oct.
Costos	14-oct.	4	18-oct.
Conclusiones	18-oct.	2	20-oct.
Codif del prototipo	20-oct.	30	19-nov.



*Diagrama de Gantt elaboración propia*

## Relevamiento

### *Relevamiento estructural*

Por tratarse de un proyecto dirigido a equipos de trabajos de una organización ficticia, no es posible localizar una ubicación física, ya que dependerá específicamente del proyecto en el que la organización se encuentre desarrollando y el equipo conformado para ella.

### *Relevamiento funcional*

#### Organigrama

No aplica por tratarse de un proyecto dirigido a autónomos y equipos de desarrollo. Asimismo, a través de la técnica de recolección de datos por medio de entrevista, se determina que no utilizan una estructura formal para los procesos funcionales, pero se logran identificar los agentes que pueden llegar a interactuar:

Analista de sistema: son los encargados de llevar el conocimiento necesario al equipo para el entendimiento y ejecución del software, usando metodologías de análisis que permitan especificar el dominio del problema.

Administrador de proyecto (Project Manager): Es un perfil que se involucra en el diseño del proyecto, es quien confecciona los tiempos, las necesidades y los objetivos como así también los riesgos y alternativas que puedan surgir, tomando decisiones.

Diseñador de Sistema: Debe definir las responsabilidades, operaciones, atributos y relaciones de una o varias clases y determinar cómo ajustarlas al ambiente de implementación. Es responsable de diseñar la base de datos definiendo las tablas, los índices, las vistas, las restricciones, triggers, los procedimientos o parámetros de almacenamiento, así como todo aquello que resulte necesario para almacenar, recuperar y borrar los objetos persistentes

Desarrolladores (Devs): Es el encargado de traducir en código los requerimientos establecidos para el sistema.



Testers: son los encargados de generar planes de pruebas y encontrar errores en la aplicación; documentan los hallazgos, planean y ejecutan ciclos de pruebas.

A continuación, se mencionan los procesos, roles y agentes involucrados en el desarrollo del proyecto:

Proceso: Gestión y Planificación del Proyecto

Roles: Project Manager

Pasos:

- Definir y presentar el proyecto al equipo.
- Organizar las tareas y asignarlas a los miembros del equipo.
- Capacitar al grupo en las metodologías y estándares a utilizar.
- Crear un modelo de ciclo de vida para el proyecto.
- Marcar objetivos a lograr.
- Definir un plan y protocolo para desarrollo de reuniones
- Construir un plan de trabajo específico que contenga diagramas Gantt y de flujo de actividades.
- Supervisar las tareas.
- Implementar cambios y soluciones que pueden surgir.

Proceso: Especificación de requisitos

Roles: Analista

Pasos:

- Entrevista al cliente, ayudándolo a identificar sus necesidades.
- Verificar si los requisitos especificados son los correctos.
- Construir el documento de requisitos de usuarios.
- Establecer una estructura básica inicial del sistema
- Definir la especificación de la arquitectura del sistema, en forma de un documento técnico comprensible.

Proceso: Diseño del sistema

Roles: Diseñador

Pasos:

- Generar diseño arquitectónico y diseño del sistema basándose en los requisitos previamente definidos.
- Generar prototipos rápidos del sistema para verificar requisitos.
- Generar documento de diseño arquitectónico (DDA).
- Seleccionar el método de almacenamiento apropiado para las estructuras de datos.
- Determinar el método apropiado para las líneas de control de ejecución.

Proceso: Desarrollo de software

Roles: Programador

Pasos:

- Traducir en código la especificación del sistema
- Determinar lenguajes posibles de usar y determinar la herramienta de desarrollo a utilizar
- Entregar un código ejecutable según las fechas especificadas
- Determinar formas de realizar un testeo
- Mantener el código eficiente y adaptable para ser unido con el código de otros programadores.
- Entregar la documentación técnica del código fuente.

Proceso: Asegurar calidad del producto

Roles: Tester

Pasos:

- participar en los procesos de especificación del sistema, previniendo errores en etapas tempranas del desarrollo.
- Revisar los documentos de requisitos de usuario y de software
- Revisar el plan de administración del proyecto
- Revisar el plan de testeo
- Revisar las políticas de control de cambios, control de errores y control de la configuración
- interactuar con el diseñador y realizar test al diseño
- realizar los tests y ajustes pertinentes
- informar sobre resultados obtenidos

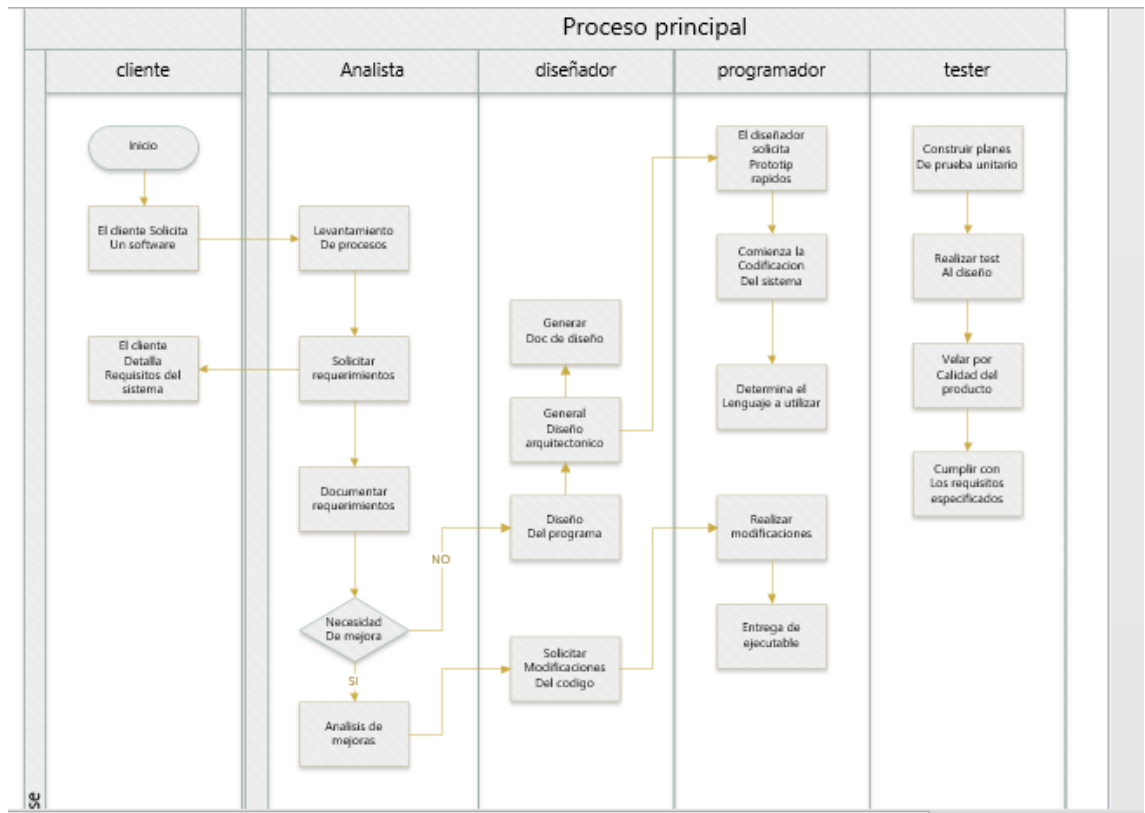
Proceso: Documentación

Roles: Documentador

Pasos:

- Determinar los tipos de documentos que se desarrollarán durante el proyecto.
- Definir su formato (.doc, .rtf, .html, .txt, .ppt, .xls, .mpp, etc.).
- Elaborar una plantilla para cada documento, en el formato definido.
- Publicar las plantillas.
- Diseñar y construir un repositorio de información compartido, donde se almacenará la documentación.
- Agregar todos los nuevos documentos generados y reemplazar los documentos que fueron modificados en el proceso de desarrollo.
- Asegurarse que los documentos mantienen el estándar de documentación definido para el proyecto antes de incluirlos en el repositorio.
- Elaborar actas de las reuniones, que luego serán utilizadas por el ingeniero de validación y verificación.
- Elaborar el manual de uso del sistema.

## Procesos de negocios



*Proceso Principal. Elaboración Propia*

## Diagnóstico y Propuesta

*Diagnóstico*

Nombre del Proceso: Especificación de requisitos	
Problemas	Causas
1. La definición del alcance no cumple con el tiempo establecido.	<ol style="list-style-type: none"> <li>1. Se realizan modificaciones que no son informadas a todos los interesados.</li> <li>2. Factores internos y/o externos que provocan demoras y no son tomados en cuenta.</li> </ol>
2. Insuficiente administración de los riesgos	<ol style="list-style-type: none"> <li>1 Mala identificación del alcance del sistema y pobre identificación de la funcionalidad requerida.</li> <li>2 Metodología de desarrollo inapropiada, falla en la definición de roles y responsabilidades, pobre planeación y control del proyecto.</li> </ol>
3. Problemas en los requisitos.	<ol style="list-style-type: none"> <li>1 Uno o más requisitos necesitan más información para definir correctamente alguna característica del sistema.</li> <li>2 Dos o más requisitos contienen información contradictoria entre sí.</li> </ol>

Nombre del Proceso: Asegurar calidad del producto	
Problemas	Causas
1 Escatimar en el control de calidad.	2 Destinar menos del 30% del desarrollo del software a las pruebas que el dpto. de calidad ejecuta.
2 diseño de test de prueba de manera errónea.	<p>1 Las funciones del sistema no funcionan de acuerdo a sus especificaciones.</p> <p>2 No se aplican test para diseñar casos de pruebas efectivos.</p> <p>3 No asegurarse que los diseñadores seleccionaran la metodología apropiada y que el producto final cumpla con los requisitos de rendimiento, diseño y verificación.</p>

Nombre del Proceso: desarrollo de software	
Problemas	Causas
1. Utilización de herramientas no adaptadas a la metodología utilizada.	1. En las actividades de diseño, el diseñador no diseñó las herramientas necesarias para esa metodología.
2 situaciones no esperadas en pequeñas modificaciones del código.	1. El programador debe interactuar con el tester para determinar una forma apropiada de construir los test y probar los programas.
3 Elección incorrecta del lenguaje de programación	1. No se exploraron los ambientes en donde el sistema puede ser desarrollado.

Nombre del Proceso: Documentación	
Problemas	Causas
1. Perfeccionar el sistema durante el tiempo.	1. Los analistas determinan nuevos requisitos.
2. Rediseñar partes del sistema	1. Existen partes del sistema que necesitan ser rediseñadas y corregidas
3. Errores en el uso del sistema	1. No hubo comunicación con el equipo de desarrollo para su reparación.

*Propuesta*

El sistema que se desarrolló, contribuyó con la planificación de actividades, mejorando la productividad, ejecución y control de tareas de un proyecto, beneficiando a los equipos de trabajo y empresas a mejorar sus productos, además mediante las pizarras personalizables, ofrecerán transparencia sobre el trabajo del equipo, proporciona herramientas de planificación para que los equipos puedan gestionar a los interesados, los presupuestos y requisitos de las funciones desde el primer día, brindando además herramientas para visualizar el estado del código en producción. Todo esto por medio de un aplicativo web, el cual puede ser accedido por una PC con acceso a internet.

## **Objetivo, Límites y Alcance Del Prototipo.**

### *Objetivo del prototipo*

Desarrollar un prototipo de sistema que permita el control y seguimiento de avances de un proyecto en desarrollo.

### *Límites del prototipo*

Desde la planificación de actividades hasta el seguimiento e informe.

### *Alcance del prototipo*

- Crear usuarios.
- Iniciar Sesión.
- Cerrar Sesión.
- Crear proyectos.
- Eliminar proyectos.
- Modificar proyectos.
- Crear pizarra de tareas.
- Eliminar pizarra de tareas.
- Modificar pizarra de tareas.
- Compartir pizarra de tareas.



- Crear tareas de proyecto.
- Crear subtareas.
- Asignar estado de tarea.
- Modificar tareas.
- Eliminar tareas.
- Modificar subtareas.
- Eliminar subtareas.
- Crear filtros para consultas.
- Monitoreo de cambios (autor, hora, fecha, acción).
- Realizar consultas en proyecto.
- Compartir filtros entre usuarios.
- Visualizar estructura e independencias de tareas en gráfico de Gantt.
- Consumir API para operaciones CRUD.
- Invitar personas a pizarras.

## **Descripción del Sistema**

### *Requerimientos funcionales*

#### *El sistema permitirá:*

RF 1: Ingresar al sistema: el sistema debe solicitar un usuario y contraseña para permitir el ingreso a la aplicación.

RF 2. Crear proyecto: El sistema debe permitir crear proyectos

RF 3. Crear tareas: El sistema debe permitir crear tareas, elegir tipo, asignarle un nombre y añadirle una descripción. Se deberá también poder asignarle la tarea a una persona.

RF 4. Crear subtareas: El sistema debe permitir crear tareas dentro la tarea principal, elegir un tipo, añadirle un nombre, una descripción y asignarle a una persona.

RF 5. Crear filtros: El sistema deberá permitir crear filtros para consultas donde se podrá filtrar las tareas por estado, tipo, proyecto.

RF6. Guardar filtros: El sistema deberá permitir guardar los filtros que se consultaron.

RF7. Compartir filtros: El sistema debe poder compartir filtros entre distintos usuarios mediante un enlace para que visualice dicha información.

RF 8. Editar tareas a través de los filtros: El sistema podrá realizar cambios en una tarea o grupo de tareas.

RF 9. Crear pizarras: el sistema podrá crear pizarras a partir de un proyecto existente, uno nuevo o desde un filtro guardado.

RF 10. Mostrar pizarras: El sistema deberá desplegar en pantalla la pizarra creada con las tareas y subtareas divididos por columnas que muestran los estados que pueden llegar a tener esas tareas.

RF 11. Transición de tareas entre columnas: El sistema debe permitir mover las tareas entre columnas dependiendo del estado en que se encuentren (en revisión, pendiente, en progreso y listas).

RF 12. Crear gráfico de Gantt: crear un gráfico de Gantt con la información de los filtros que permite visualizar la estructura de nuestro proyecto y la independencia de las tareas.

### *Requerimientos no funcionales.*

El sistema deberá cubrir los siguientes aspectos:

#### RNF 1: Usabilidad

- Intuitivo.
- Poseer una guía que permita entender y aprender sus funciones.
- Poseer mensajes de alerta, avisos y errores.

#### RNF 2: Portabilidad

- Que se adapte de forma efectiva a PC escritorio.
- Capacidad de ser transferido de un entorno a otro (hardware, organización o software)

### RNF 3: Seguridad

- Cada usuario deberá ingresar al aplicativo usando correo electrónico y contraseña, la cual deberá contener como mínimo 6 caracteres alfanuméricos.

### RNF 4: Legalidad

- Se deberá cumplir con lo mencionado en la Ley 25.326 sobre protección de datos personales.

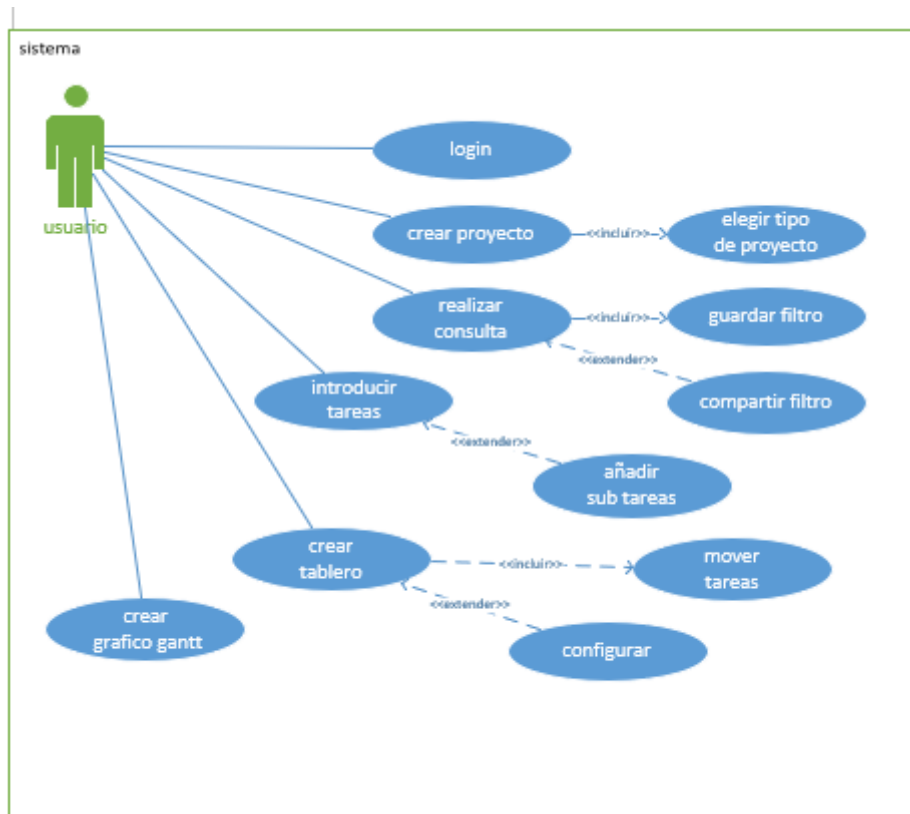
### RNF 5: Mantenibilidad

- Capacidad del producto de ser modificado.

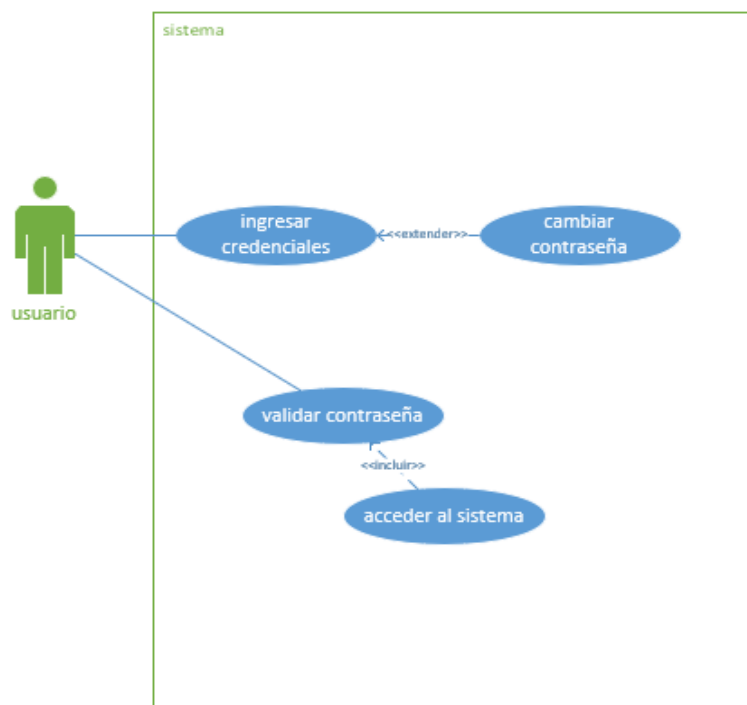
### RNF 6: Eficiencia

- Capacidad de proporcionar un desempeño apropiado en relación con la cantidad de recursos utilizados.

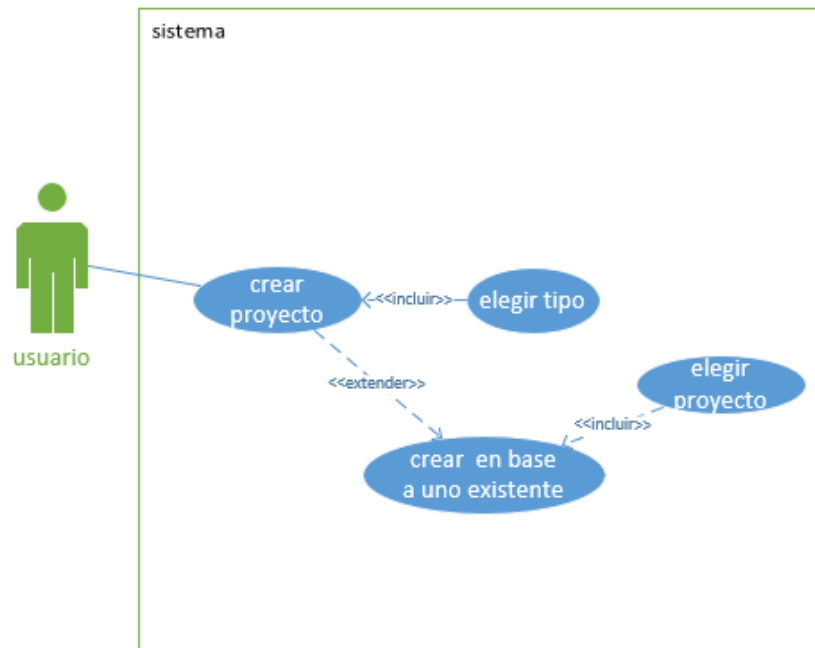
### *Diagrama de casos de usos*



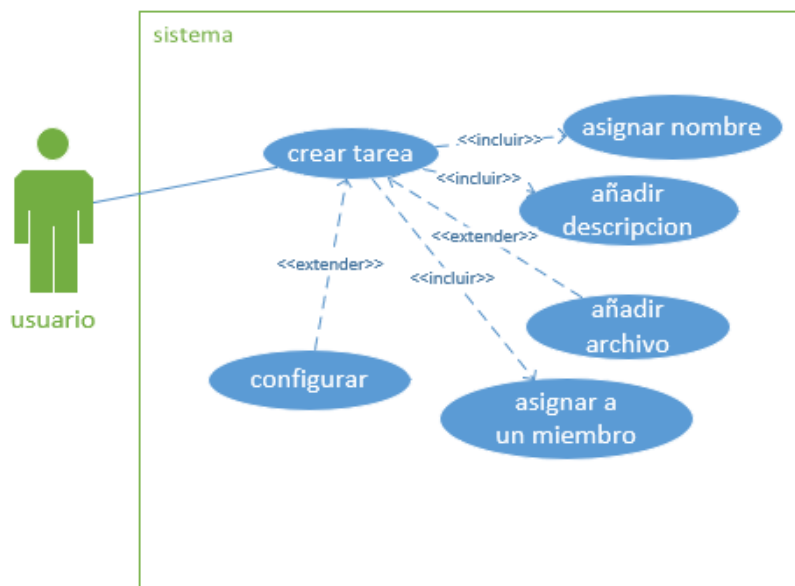
Caso de uso general



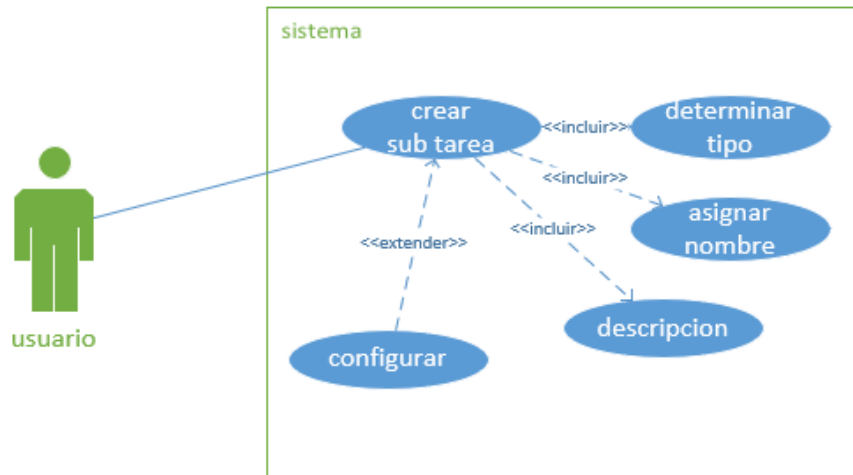
Caso de uso ingresar al sistema



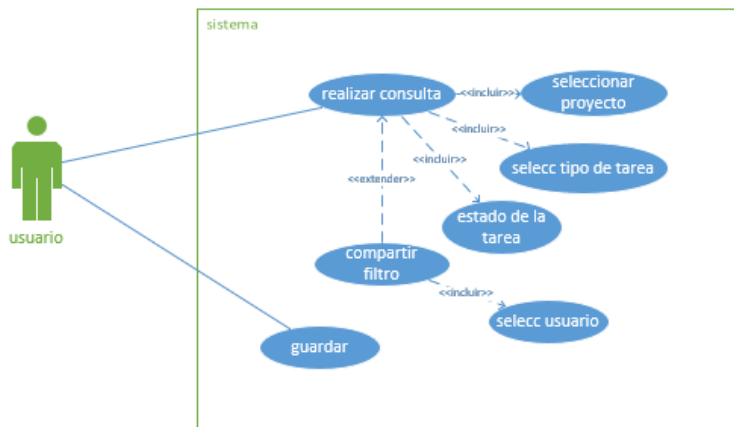
### Caso de uso Crear un nuevo Proyecto



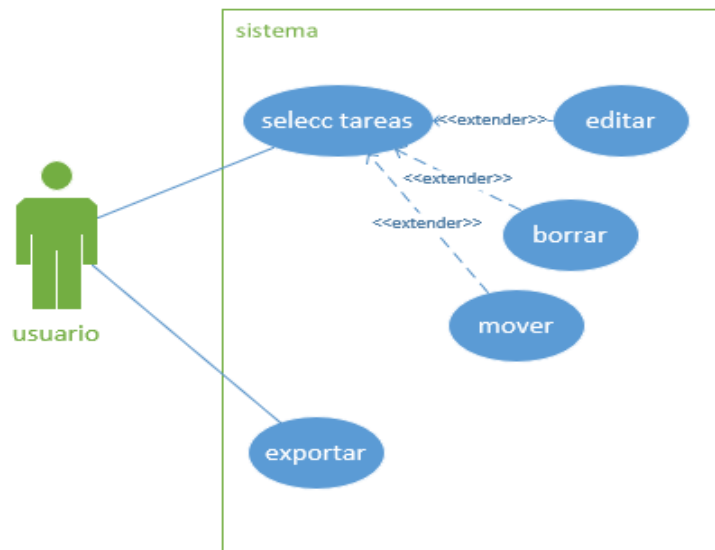
## Caso de uso crear una tarea



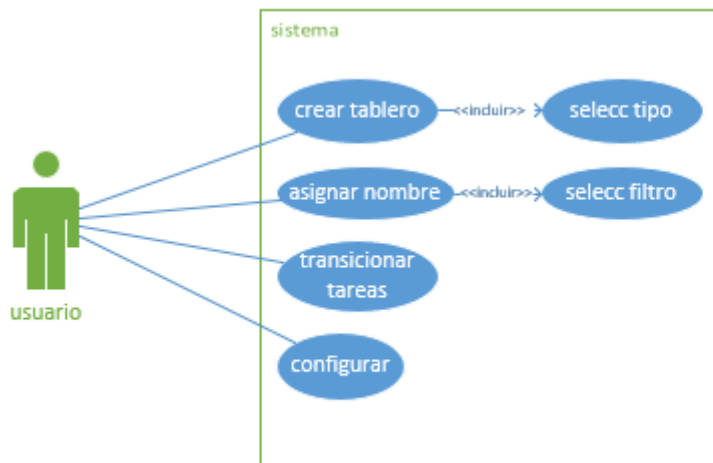
## Caso de uso crear una sub tarea



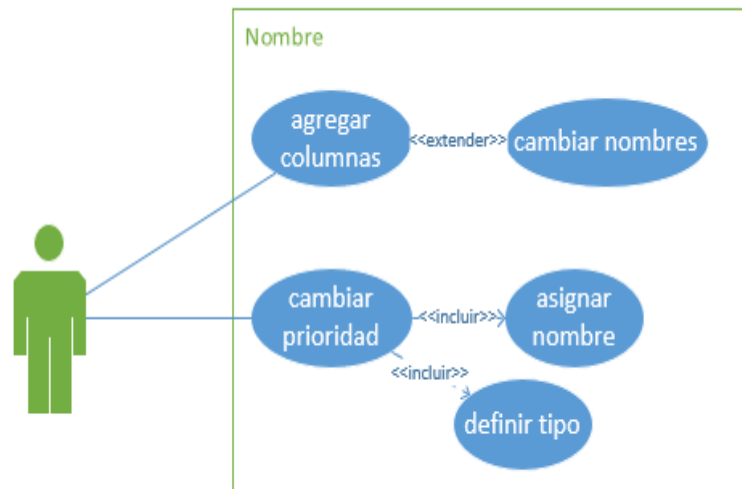
## Caso de uso crear filtros



Caso de uso Modificar tareas



Caso de uso Crear una pizarra



Caso de uso Configuración de la pizarra



Caso de uso pizarra de Gantt



*Descripción de casos de uso*

RF 2	Ingresar al sistema	
Versión	1.2	
Objetivos asociados	Ingresar al sistema. Registrar usuario. Modificar usuario.	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario desee ingresar para comenzar a utilizar la aplicación.	
Precondición	El usuario debe estar dado de alta en el sistema.	
Secuencia normal	Paso	Acción
	1	Ingresa a la sección de login.
	2	Completa los campos de usuario y contraseña.
	3	Se validan los datos.
	4	Accede al sistema.
	5	Fin.
Pos condición	El usuario puede comenzar a utilizar su aplicación.	
Curso alternativo	Paso	Acción
	3	Si la validación es incorrecta el sistema consultará si desea registrar un nuevo usuario.
	3.a	El usuario acepta generar un nuevo usuario.
	3.b	El usuario completa los datos solicitados para darse de alta en el sistema.

	3.c	El sistema confirma el alta.
	3.d	El usuario accede al sistema.
	3.e	Fin.
Frecuencia esperada	5 veces por día.	
Importancia	Importancia media.	
Comentarios	La cantidad de accesos por día dependerá del usuario y su necesidad de gestionar o realizar consultas.	

RF 2	Crear un Nuevo Proyecto	
Versión	1.2	
Objetivos asociados	Crear un proyecto de trabajo	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario cree un nuevo proyecto	
Precondición	El usuario debe estar dado de alta en el sistema.	
Secuencia normal	aso	Acción
		ir a la barra de menú Proyecto>Crear Proyecto.
		seleccionar tipo de proyecto deseado.

		El usuario deberá asignar un nombre al proyecto
		Presionar botón crear.
		fin
Post condición	El usuario puede comenzar a agregar tareas al proyecto.	
Curso alternativo	aso	Acción
	3	Si el usuario desea crear un proyecto en base a uno ya existente.
	2	ir a la barra de menú Proyecto>Crear Proyecto
	3	el usuario deberá seleccionar la opción “crear con configuración compartida”
	4	se generará una ventana donde el usuario elegirá el proyecto a utilizar a través de una pestaña de despliegue.
	5	El usuario asignará un nombre al proyecto, automáticamente se genera una clave única al proyecto que servirá de prefijo para identificar unívocamente cada proyecto.
	6	fin

Frecuencia esperada	5 veces por día.
Importancia	Importancia media.
Comentarios	La cantidad de accesos por día dependerá del usuario y su necesidad de crear proyectos.

RF 2	Crear una tarea	
Versión	1.2	
Objetivos asociados	Crear una tarea específica.	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario desee comenzar a añadir tareas.	
Precondición	El usuario debe estar logueado en el sistema.	
Secuencia normal	Paso	Acción
	1	Ingresa a la página principal.
	2	Presionar el botón “crear” que se encuentra en la barra de menú.
	3	Seleccionamos el proyecto con el que estamos trabajando.
	4	Determinamos el tipo de tarea que será.

	5	seleccionamos tipo de tarea “tarea principal”
	6	Asignamos un nombre.
	7	Añadimos una descripción.
	8	Se le asignan los usuarios.
Post condición	El usuario añade una nueva tarea.	
Curso alternativo		
Frecuencia esperada	5 veces por día.	
Importancia	Importancia media.	
Comentarios	La cantidad de tareas agregadas por día dependerá del usuario y lo que demande el proyecto.	

RF 1	Crear una sub tarea
Versión	1.3
Objetivos asociados	Crear sub tareas dentro de una tarea principal.
Descripción	El sistema se comportará de la siguiente manera cuando el usuario decida agregar sub tareas.
Precondición	El usuario debe estar registrado en el sistema.

	Paso	Acción
Secuencia normal	1	El usuario ingresa a una tarea.
	2	Selecciona el proyecto que desea trabajar.
	3	Se le asigna un nombre a esa sub tarea
	4	Seleccionamos el tipo de tarea que será “sub tarea”
	5	Se le añade una descripción
	5	Luego el usuario presiona el botón crear.
	6	El sistema procesara esa acción.
	7	Se despliega un msj que la tarea se añadió con éxito.
	8	Fin.
Pos condición	El usuario puede visualizar y gestionar las sub tareas añadidas.	
Curso alternativo		
Frecuencia esperada	2 veces por día.	
Importancia	Importancia alta.	
Comentarios	La cantidad de tareas y sub tareas añadidas dependerá del equipo de trabajo y el proyecto.	

RF 1	Crear filtros
Versión	1.3
Objetivos asociados	Hacer consultas en un proyecto Filtrar tareas por estado Asignar filtro a distintos miembros

	Compartir enlace del filtro entre compañeros.	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario cree un filtro.	
Precondición	El usuario debe estar logueado en el sistema. Debe haber actividades registradas	
Secuencia normal	Paso	Acción
	1	El usuario se dirige a la barra de menú donde selecciona el botón “realizar búsqueda”
	2	Filtramos por proyecto, tipo de tarea a consultar, estado.
	3	El usuario procede a guardar los cambios realizados.
	4	El sistema procesará la respuesta del usuario.
	5	El usuario luego podrá compartir ese enlace al filtro dirigiéndose al apartado de “detalles” en la barra de menú superior.
	6	El sistema desplegará una pestaña y seleccionamos “permisos”
	7	El usuario asignará un nombre en el campo “name”
	8	Luego el usuario marcará a quienes podrán acceder al enlace
	9	El usuario aprieta la opción “guardar”
	10	Fin.
Post condición	El usuario puede visualizar y gestionar el los filtros creados.	
Curso alternativo		
Importancia	Importancia alta.	

Comentarios	La cantidad de accesos por día dependerá del usuario y su necesidad de gestionar los filtros.
-------------	---

RF 1	Modificar tareas	
Versión	1.3	
Objetivos asociados	Realizar cambios en una o varias tareas a través de un filtro	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario comience a gestionar las tareas.	
Precondición	El usuario debe estar logueado en el sistema.	
Secuencia normal	Paso	Acción
	1	El usuario ingresa a la opción “herramientas” desde un filtro determinado.
	2	Selecciona las tareas que desea modificar.
	3	Luego el usuario presionara la opción de “siguiente”
	4	El sistema mostrara una serie de opciones para seleccionar.
	5	El usuario decidirá si desea eliminar, mover, editar las tareas.
	6	Luego el usuario presiona siguiente.
Post condición	El usuario podrá visualizar los cambios realizados.	
Curso alternativo		
Frecuencia esperada	3 veces por día.	
Importancia	Importancia baja.	
Comentarios	La cantidad de modificaciones por día dependerá de los cambios que surjan en el desarrollo del proyecto.	



RF 1	Crear una pizarra.	
Versión	1.3	
Objetivos asociados	<p>Crear una pizarra de actividades.</p> <p>Registrar tareas asociadas al proyecto.</p> <p>visualizar estado de tareas.</p> <p>Transicionar las actividades entre columnas</p>	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario desee crear un panel de actividades.	
Precondición	El usuario debe estar logueado en el sistema.	
Secuencia normal	Paso	Acción
	1	El usuario ingresa a la opción crear pizarra.
	2	Selecciona la el tipo de pizarra que desea crear.
	3	Presiona el botón siguiente
	4	El sistema procesará la respuesta del usuario.
	5	El sistema mostrara una ventana con opciones.
	6	El usuario elegirá si desea trabajar con un nuevo proyecto, un proyecto existente o un filtro guardado.
	7	Añadirá un nombre a la pizarra.
	8	Asignamos un filtro y presionamos “crear pizarra.”
	9	El sistema procesará la respuesta del usuario.

	10	El usuario ingresa a consultar emociones.
	11	El sistema mostrara los estados de ánimo registrados.
	12	Fin.
Post condición	El usuario puede visualizar y gestionar el seguimiento de su agenda visual.	
Curso alternativo	Paso	Acción
	5	Si el usuario desea puede confirmar las actividades cumplidas.
	5.a	El sistema preguntará si desea guardar los cambios.
	5.b	El sistema registrará la respuesta del usuario.
	5.c	Fin.
	7	Si el usuario desea puede asociar el estado anímico a una actividad registrada en la agenda.
	7.a	El sistema preguntará si desea guardar los cambios
	7.b	El sistema registrará la respuesta del usuario.
	7.d	Fin.
Frecuencia esperada	5 veces por día.	
Importancia	Importancia alta.	
Comentarios	La cantidad de accesos por día dependerá del usuario y su necesidad de gestionar o realizar consultas.	

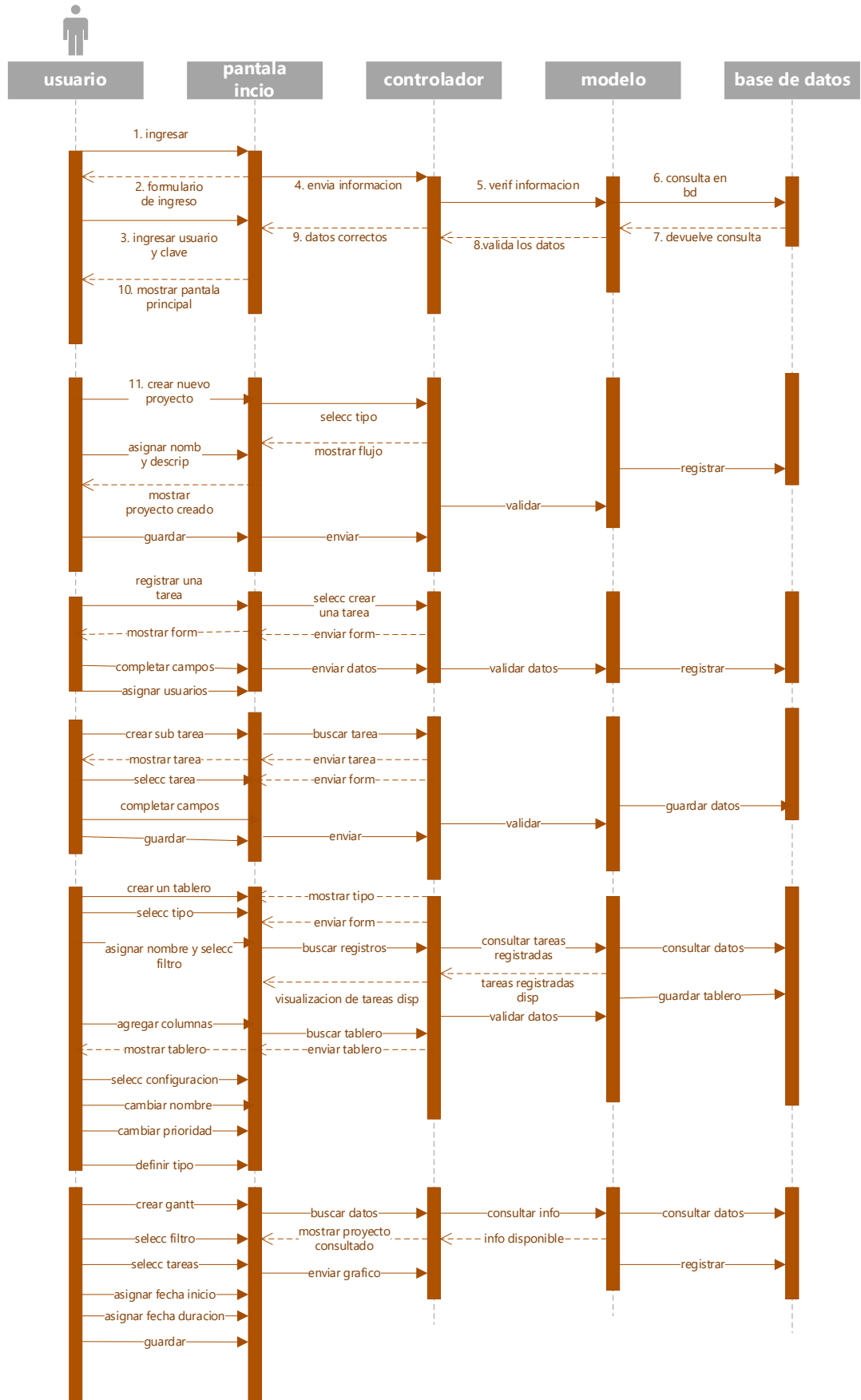
RF 1	Configuración de la pizarra.	
Versión	1.3	
Objetivos asociados	Adaptar pizarra a las necesidades del usuario.	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario desee realizar cambios en el panel de actividades.	
Precondición	El usuario debe estar logueado en el sistema.  El sistema debe tener creado una pizarra.	
Secuencia normal	Paso	Acción
	1	El usuario ingresa a la pizarra.
	2	Se dirige al botón herramientas en la esquina superior derecha.
	3	Al costado izquierdo podrá visualizar una columna con distintas opciones.
	4	El usuario podrá elegir que opción elegir.
	5	El sistema mostrara una ventana con opciones.
	6	Si el usuario elige la opción columnas podrá agregar, quitar, cambiar estados o pasar tareas entre columnas.
7	Fin.	

Pos condición	El usuario puede visualizar y gestionar las pizarras creadas.	
Curso alternativo		
Frecuencia esperada	2 veces por día.	
Importancia	Importancia media.	
Comentarios		

RF 1	Crear Gantt	
Versión	1.3	
Objetivos asociados	Crear un diagrama que permita visualizar la duración de las actividades de un proyecto.	
Descripción	El sistema se comportará de la siguiente manera cuando el usuario desee crear un diagrama gráfico con la información de las pizarras registradas en el sistema.	
Precondición	El usuario debe estar logueado en el sistema.  El sistema debe tener creado una pizarra.	
Secuencia normal	Paso	Acción
	1	El usuario se dirige a la barra de menú seleccionando la opción grafico de Gantt.
	2	El sistema mostrará la estructura de nuestro proyecto y la dependencia

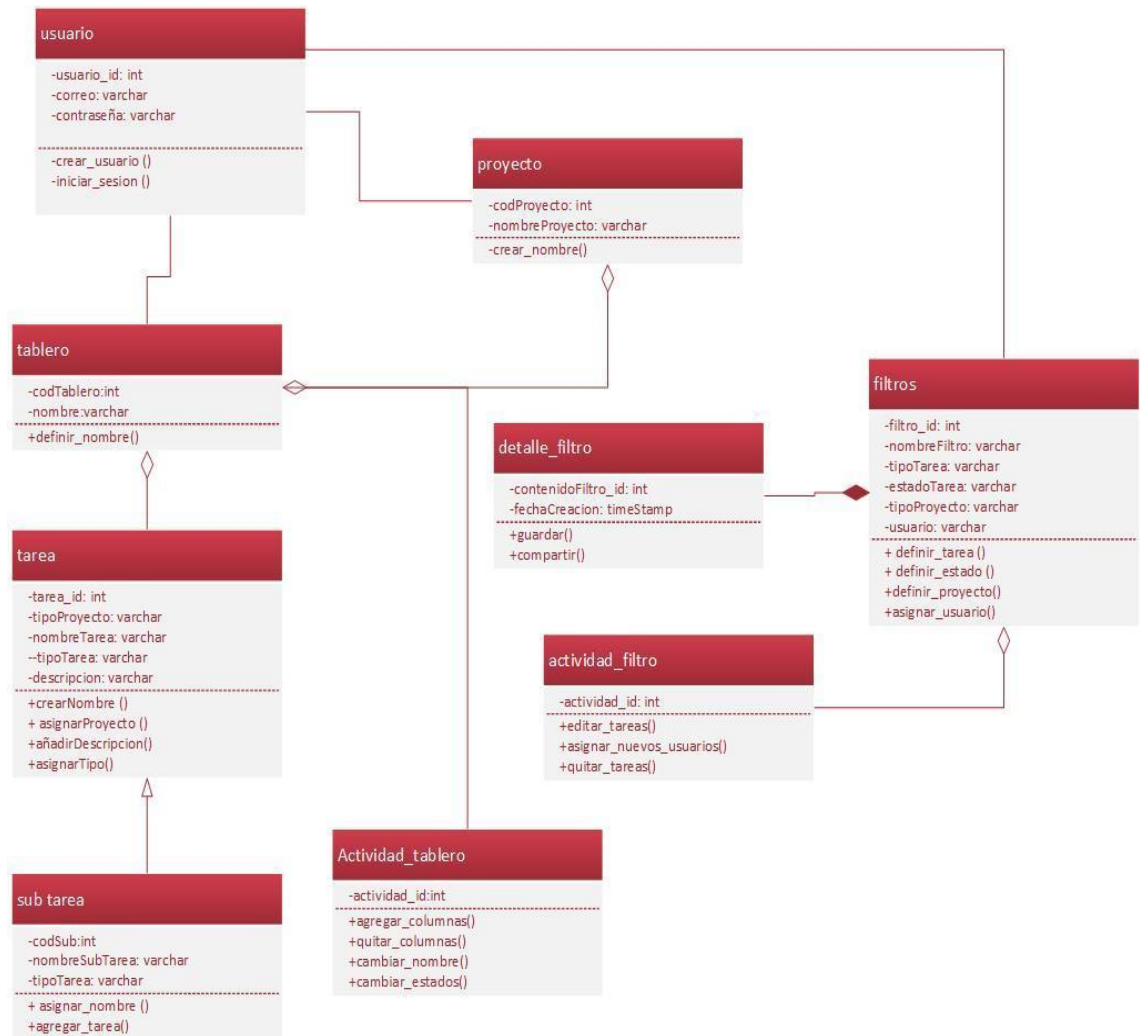
		entre las tareas en forma de calendario.
	3	El usuario comenzará a aplicar la fecha de inicio de cada tarea seleccionando dicha información en una pestaña que se ira desplegando.
	4	El usuario determinará en otra columna la fecha de duración de cada actividad de manera manual o desplegando una pestaña con la información necesaria.
	5	Fin.
Pos condición	El usuario puede visualizar y gestionar los los gráficos creados.	
Curso alternativo		
Frecuencia esperada	2 veces por día.	
Importancia	Importancia media.	
Comentarios		

*Diagrama de secuencia*

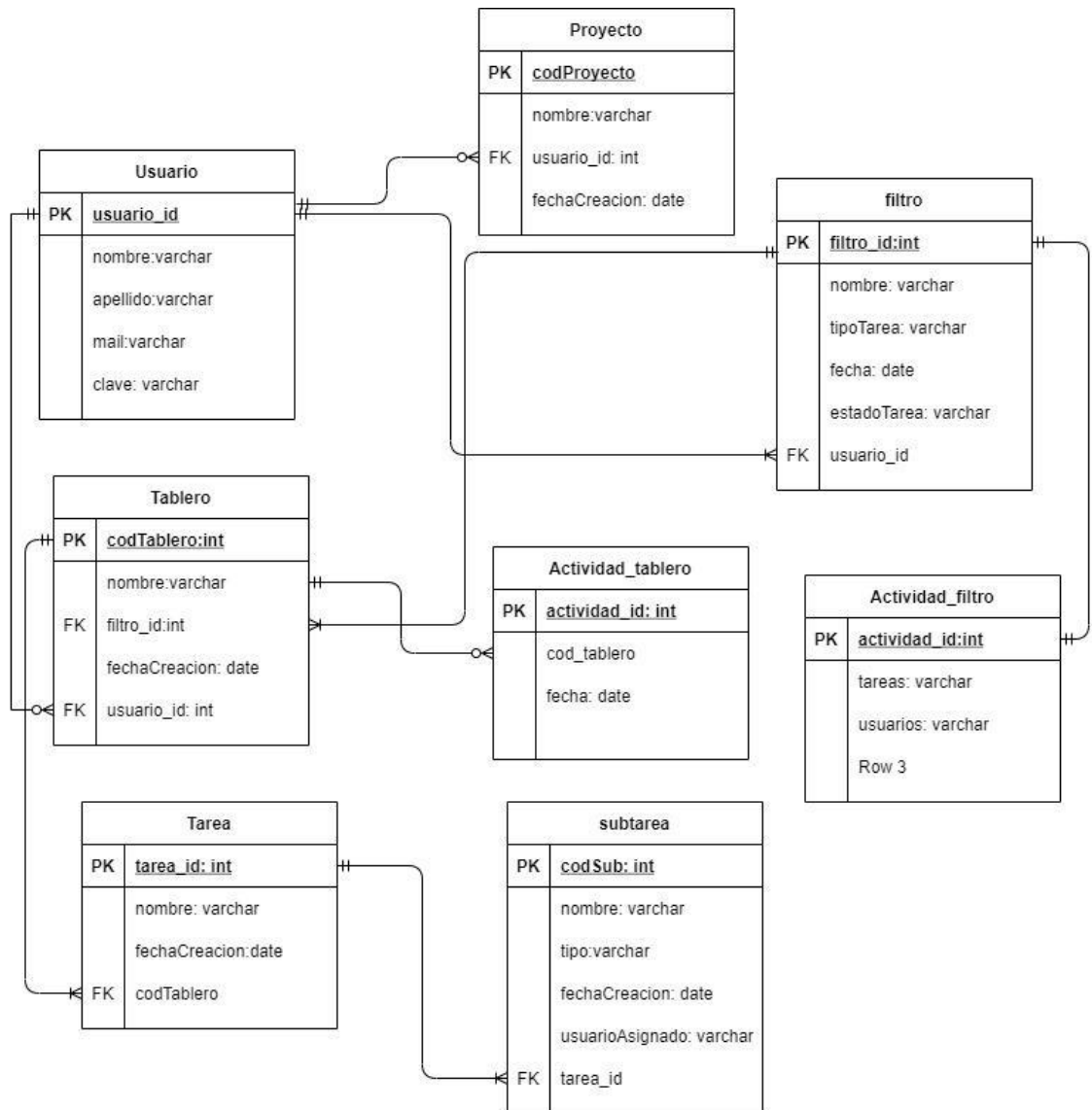


## Estructura de datos

### Diagrama de clases



*Diagrama entidad relación (DER)*



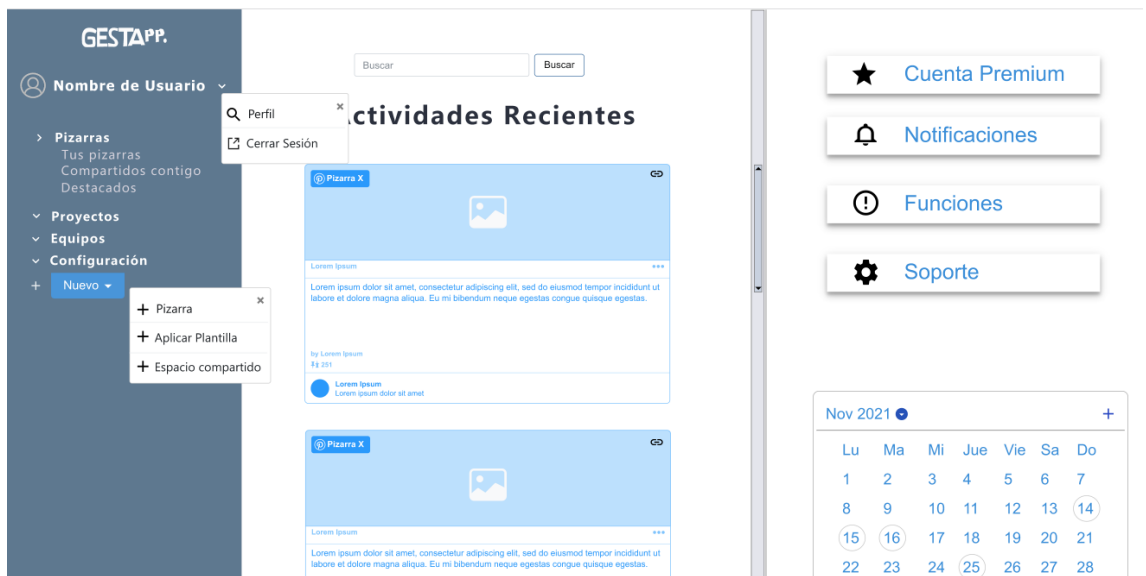
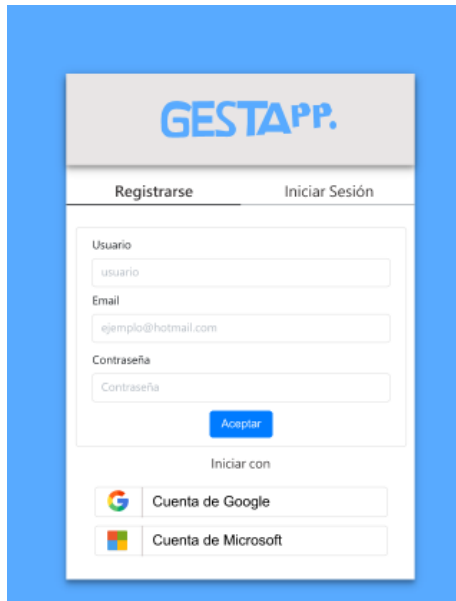
*Prototipo de interfaz de pantalla*

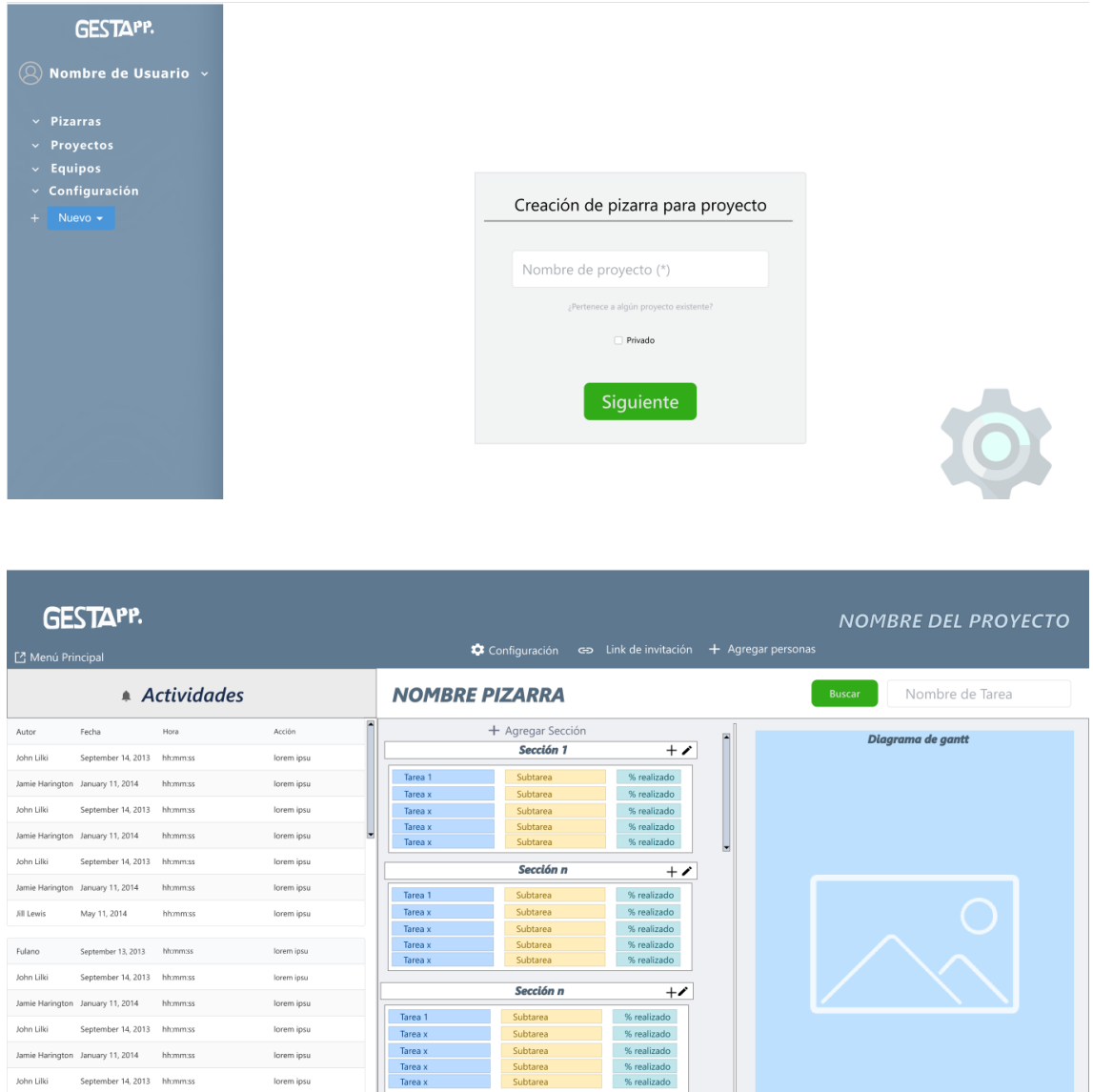


## Pantalla Bienvenida del prototipo



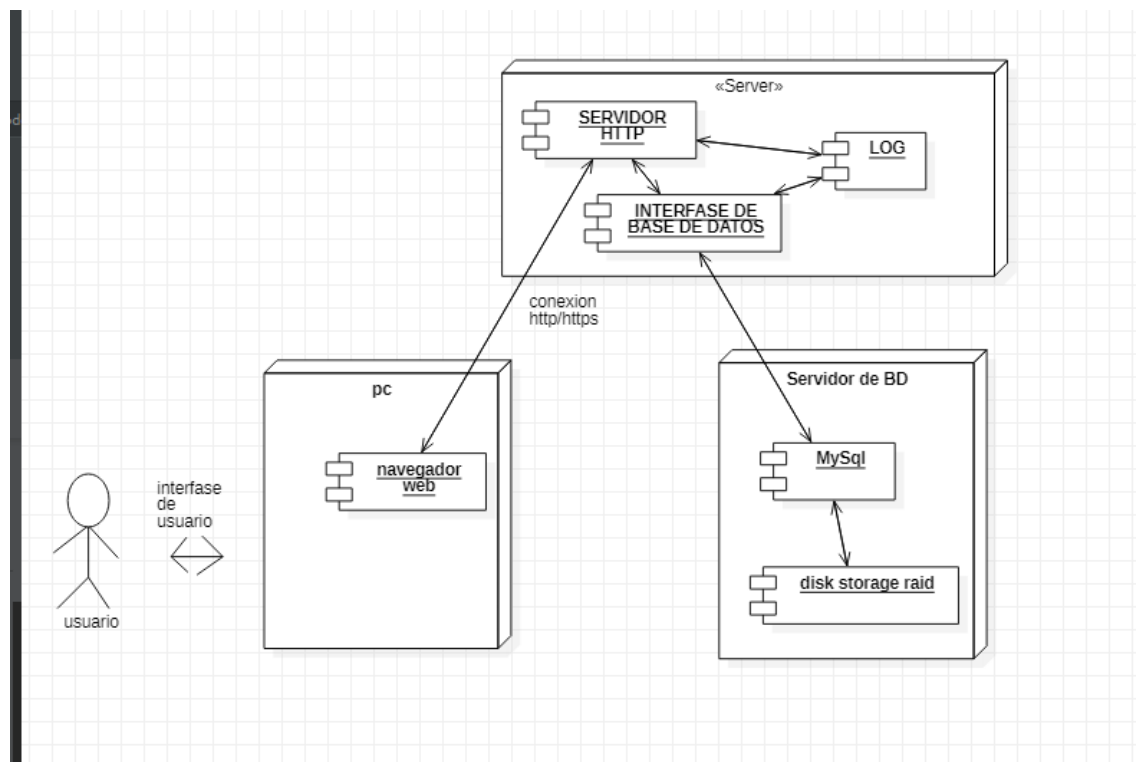
## Pantalla principal de ingreso





Prototipo de interfaz de pantalla. Elaboración propia.

### Diagrama de despliegue



## Seguridad

### Acceso a la aplicación

La plataforma desarrollada se comunicará mediante el protocolo HyperText Transfer Protocol Secure (HTTPS) y utilizará el certificado Transport Layer Security (TLS) que deriva del protocolo Secure Sockets Layer (SSL) y es la tecnología estándar para mantener segura una conexión de internet. TLS permite y garantiza el intercambio de datos en un entorno securizado y privado entre dos entes, el usuario y el servidor. Por lo tanto, se lleva a cabo un canal seguro y cifrado entre cliente y servidor en donde

se negocia la criptografía del mensaje, se autentifican las claves del cifrado y se realiza una transmisión segura. (swhosting, s.f.)

Para acceder a la aplicación si el usuario no posee una cuenta, debe registrarse utilizando un nombre de usuario, cuenta de correo electrónico y una contraseña que debe cumplir con los siguientes requisitos:

- Tener como mínimo, 8 caracteres.
- Estar compuesta por la combinación de letras y números.
- Poseer al menos, una letra mayúscula.

En caso que el usuario posea una cuenta, debe ingresar a la aplicación utilizando el correo electrónico asociado a la misma y su correspondiente contraseña, la cual:

- Si se ingresa erróneamente 5 veces se bloqueará el acceso, enviando una alerta por correo.

También se contempla que, tanto para el registro como el inicio de sesión, el usuario tenga la posibilidad de ingresar a la aplicación mediante cuentas de Google o Microsoft.

La aplicación maneja varios perfiles dependiendo de los roles y permisos que se le asignen:

Administrador	Usuario A
Es el usuario con más acceso a funciones ya que permite:	Solo permite:
<ul style="list-style-type: none"> <li>● Generar tareas</li> <li>● Crear proyectos</li> <li>● Asignar roles</li> <li>● Asignar permisos</li> <li>● Modificarlas</li> <li>● Eliminar</li> <li>● Gestionar perfil</li> <li>● Administrar sprints</li> </ul>	<ul style="list-style-type: none"> <li>● modificar tareas</li> <li>● buscar un proyecto</li> <li>● agregar comentarios</li> <li>● examinar proyectos</li> <li>● registrar cambios</li> <li>● agregar tareas a la pizarra</li> </ul>

Tabla tipos de usuarios. Elaboración propia.

### *Respaldo de información*

Servicios en la nube: Una copia de los datos del servidor se almacenarán en un servidor remoto gratuito. Aquí se tendrán los datos originales y el respaldo de estos datos. Esto nos permitirá mantener los archivos y datos disponibles en caso de que nuestro sistema falle. Estas copias se actualizarán semanalmente. En cuanto a seguridad, en la nube se aplica una amplia gama de políticas, tecnologías y formas de control, destinadas a proteger los datos. Se utilizará el cifrado del lado del servidor, que es el método que utilizan la mayoría de servicios de almacenamiento de archivos en la nube. Esto se refiere a que la información llega al servidor sin cifrar, y allí es cifrada. La transferencia de los archivos se realizará a través de una conexión segura (HTTPS/SSL).

Servidor local: Se realizarán copias en dispositivos de almacenamiento físicos como discos duros. Esto nos traerá como ventaja la accesibilidad inmediata y rápida al no depender de una conexión a internet. Esta información estará segura ya que sólo es accesible para los administradores que puedan acceder a dichos almacenamientos locales siempre que se lo requiera o con periodicidad a estimar y establecer en políticas preventivas y prohibitivas de la organización.

## **Análisis de costo**

### *Recursos humanos*

Para llevar adelante este proyecto se definió una estructura de recursos humanos conformada por un líder de proyecto, dos programadores Full Stack, uno para el desarrollo Front-end y el segundo para el desarrollo Back-End. Para la confección de la base de datos por tratarse de uno de los procesos centrales de la estructura del sistema se optó por contratar los servicios de un administrador de base de datos.

### Análisis de costo de desarrollo

<b>Rol</b>	<b>Costo hora</b>	<b>Cantidad de horas</b>	<b>Sub total</b>
Líder de proyecto	\$ 4.420	200	\$ 884.000
Programador front - End	\$ 3.490	200	\$ 698.000
Programador Back- End	\$ 3.490	200	\$ 698.000
Administrador de Base de Datos	\$ 4.000	100	\$ 400.000
		<b>TOTAL</b>	<b>\$ 2.680.000</b>

Tabla de costo de Recursos Humanos. Elaboración propia

Los valores de costo hora fueron tomados de la página del Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba el día 19/10/2021 y todos los valores están expresados en pesos argentinos. (CPCIPC Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba, s.f.)

#### *Hardware y software*

Se presentan a continuación las siguientes opciones de costo operativo:

#### Análisis de costo operativo opción 1. Alquiler del servicio

<b>Recurso</b>	<b>descripción</b>	<b>fuelle</b>	<b>Sub total</b>	<b>Sub total mensual</b>
procesador	Ryzen 5 1600	<a href="https://compragamer.com/?criterio=ryzen%205%203600&amp;x=0&amp;y">https://compragamer.com/?criterio=ryzen%205%203600&amp;x=0&amp;y</a>	\$ 20.000	n/a

		=0&seccion=3&nro_max=50		
memoria	16gb ddr4 3200mhz	https://compragamer.com/?criterio=16gb%20&x=0&y=0&seccion=3&nro_max=50	\$ 9.600	n/a
Almacenamiento principal	Disco rígido Seagate 1TB 64mb cache 6gb/s	https://compragamer.com/?criterio=1tb&x=0&y=0&seccion=3&nro_max=50	\$4900	n/a
Sistema operativo	Windows 10 pro	https://compragamer.com/?criterio=sistema%20operativo&x=0&y=0&seccion=3&nro_max=50	\$15.420	n/a
Ancho de banda	100mbps		n/a	\$1.800
Servidor	professional	https://www.hostinger.com.ar/precios	n/a	\$7000
		TOTAL	\$ 49.920	\$ 8.800

Tabla opción 1. Alquiler del servicio. Elaboración propia

#### Características específicas del servidor

Recurso	Descripción
Procesador	8 núcleos
Memoria	8gb de ram



Almacenamiento principal	160gb
Almacenamiento secundario	n/a
Sistema operativo	CentOS7
Ancho de banda	8tb
fuelle	<a href="https://www.hostinger.com.ar/precios">https://www.hostinger.com.ar/precios</a>
Costo mensual	\$ 7000

Opción 2 compra todo el Hardware:

Recurso	Cantidad	Fuente	Costo	Sub total mensual
Servidor Dell Poweredge T140 Xeon E-2224-4C 64GB 2TB	1	Servidores en Oferta en Oportutek.com   Dell, HP, Lenovo y más	\$ 291.499	n/a
Disco Rígido WD 1TB BLUE 64MB SATA 6.0GB/s	1	<a href="https://compragamer.com/index.php?criterio=disco%201tb&amp;x=0&amp;y=0&amp;seccion=3&amp;nro_max=50">https://compragamer.com/index.php?criterio=disco%201tb&amp;x=0&amp;y=0&amp;seccion=3&amp;nro_max=50</a>	\$ 5120	n/a
Ups 800va Apc Back- Ups Bx800ci	1	<a href="https://oportutek.com/search?type=product&amp;q=ups+800va+Apc+Back-ups+Bx800ci+">https://oportutek.com/search?type=product&amp;q=ups+800va+Apc+Back-ups+Bx800ci+</a>	\$29000	n/a

Mini Rack Mural 12u 19 Armado. Con Fondo Gabinete Redes Serv	1	<a href="http://wp.shopix.com.ar/enventa-mini-rack-mural-12u-19-armado-con-fondo-gabinete-redes-serv_SPA823755820">http://wp.shopix.com.ar/enventa-mini-rack-mural-12u-19-armado-con-fondo-gabinete-redes-serv_SPA823755820</a>	\$17.800	n/a
Eset® Nod32 Antivirus * Tienda Oficial Eset* 3 Pc - 1 Año	1	<a href="https://ar.tienda.eset-la.com/checkout/cart/?cat=hogar">https://ar.tienda.eset-la.com/checkout/cart/?cat=hogar</a>	\$ 5.093	n/a
Notebook Dell 3505 AMD Ryzen 5 3450U 16GB de RAM 1TB HDD 256GB SSD, AMD Radeon Vega 8 60 Hz	1	<a href="https://shopix.com.ar/enventa-notebook-dell-inspiron-3505-gris-15-6-amd-ryzen-5-3450u-16gb-de-ram-1tb-hdd-256gb-ssd-amd-radeon-vega-8-1366x768px-windows-10-home_SPA913505129">https://shopix.com.ar/enventa-notebook-dell-inspiron-3505-gris-15-6-amd-ryzen-5-3450u-16gb-de-ram-1tb-hdd-256gb-ssd-amd-radeon-vega-8-1366x768px-windows-10-home_SPA913505129</a>	\$ 149.999	n/a
Servicio internet cablevisión 100mb	12	<a href="https://www.personal.com.ar/ofertas">https://www.personal.com.ar/ofertas</a>	n/a	\$1.800
TOTAL			\$ 465.403	\$ 1.800

Tabla compra de todo el hardware. Elaboración propia

El costo total de recursos humanos, hardware y software es el siguiente:

opciones	costo
----------	-------

Costo total comprando todo el hardware	\$ 3.145.000
Costo total alquilando servicio	\$ 2.738.720

Se elige la opción 1A por tratarse de un servicio de alquiler de servidor que permite la opción de backup en un servidor diferente de donde se hospeda el sistema principal, además que ofrece una capacidad de almacenamiento de 1TB en comparación de los 160GB del proveedor 1B. Se descarta la posibilidad de montar un servidor propio por no disponer de un lugar físico. Y por último en referencia al software necesario para programar el aplicativo se utiliza software libre el cual no requiere de un costo por licencia.

### Análisis de Riesgos

Área	Riesgo	Causa
Planificación	Planificación muy optimista Un retraso de una tarea provoca el incumplimiento de las demás.	Error en las estimaciones Muchas tareas dependen de una sola
Organización y Gestión	Costos elevados	Presupuesto establecido en la etapa de planificación es poco realista. No se tiene en cuenta una inflación.
Usuarios finales	Usuarios insatisfechos con el producto final. Re diseñar requisitos. El usuario decide realizar cambios en los requisitos.	Falta de claridad por parte del equipo sobre las necesidades del cliente durante el proyecto. Poca claridad en los requerimientos.

	Proporcionar componentes de baja calidad por lo que se traduce a mayor tiempo para mejorar la calidad.	Falta de reuniones informativas y fallas en la comunicación con el equipo para mostrar avances.
Recursos humanos	<p>El equipo no alcanza el nivel de desempeño esperado.</p> <p>No se asigna la cantidad de personal suficiente al proyecto.</p> <p>Problemas de los miembros del equipo para trabajar de manera conjunta</p> <p>Módulos propensos a generar errores</p>	<p>Falta de evaluaciones de desempeño.</p> <p>Falta de políticas de incentivos.</p> <p>Fracaso en la fase de selección y preselección del personal.</p> <p>Poco compromiso del líder y escasa experiencia</p> <p>Necesitan más trabajo de comprobación, diseño e implementación</p>
proceso	Entregas fuera de plazos establecidos	Falta de seguimiento y control exacto del proceso de desarrollo del proyecto.

Para valorizar los mencionados riesgos se toma como referencia la siguiente matriz de riesgo:

				Impacto				
				Muy baja	Baja	Media	Alta	Muy alta
				1	2	3	4	5
	Muy alta	90%	0,9	0,9	1,8	2,7	3,6	4,5

Probabilidad de ocurrencia	Alta	70%	0,7	0,7	1,4	2,1	2,8	3,5
	Media	50%	0,5	0,5	1,0	1,5	2,0	2,5
	Baja	30%	0,3	0,3	0,6	0,9	1,2	1,5
	Muy baja	10%	0,1	0,1	0,2	0,3	0,4	0,5

Tabla de análisis cuantitativo de riesgos:

Riesgo	Probabilidad de ocurrencia	impacto
Planificación muy optimista	75%	4
Retraso en una tarea provoca el incumplimiento de las demás	68%	3
Costos elevados	93%	4
Usuarios insatisfechos con el producto final. Re diseño de requisitos	67%	3
El usuario decide realizar cambios en los requisitos	63%	3
Proporcionar componentes de baja calidad por lo que se traduce a mayor tiempo para mejorar la calidad	44%	4
El equipo no alcanza el nivel de desempeño esperado.	60%	2
No se asigna la cantidad de personal suficiente al proyecto.	10%	1
Problemas de los miembros del equipo para trabajar de manera conjunta	25%	3
Entregas fuera de plazos establecidos	35%	3

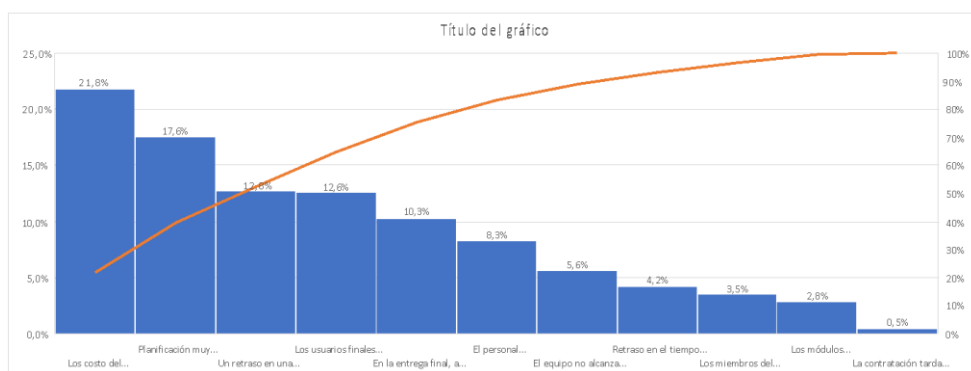
Análisis cuantitativo de riesgos. Elaboración propia.

Tabla de análisis cuantitativo y grado de exposición:

<b>Riesgo</b>	<b>Probabilidad de ocurrencia</b>	<b>impacto</b>	<b>Grado de exposición</b>	<b>porcentaje</b>	<b>% acumulado</b>
Costos elevados	93%	5	4.65	21,8%	21,8%
Planificación muy optimista	75%	5	3,75	17,6%	39,4%
Retraso en una tarea provoca el incumplimiento de las demás	68%	4	2,72	12,8%	52,2%
Usuarios insatisfechos con producto final. Re diseño de requisitos	67%	4	2.68	12.6%	64.8%
El usuario decide realizar cambios en los requisitos	55%	4	2.2	10.3%	75.1%
Proporcionar componentes de baja calidad por lo que se traduce a mayor tiempo para mejorar la calidad	44%	4	1.76	8.3%	83.3%
El equipo no alcanza el nivel de desempeño esperado.	60%	2	1.2	5.6%	89.0%
No se asigna la cantidad de personal suficiente al proyecto.	30%	3	0.9	4.2%	93.2%

Problemas de los miembros del equipo para trabajar de manera conjunta	25%	3	0.75	3.5%	96.7
Módulos propensos a recibir errores	30%	2	0.6	2.8	99.5%
Entregas fuera de plazos establecidos	10%	1	0.1	0.5%	100%

Análisis cuantitativo de riesgos acumulados. Elaboración propia



Se presentan los riesgos más relevantes con su respectivo plan de contingencia:

Riesgo	Contingencia
Costos elevados	Planificar de manera adecuada implementando estrategias que se ajusten a la inflación.
Planificación muy optimista	Realizar una planificación con mayor nivel de detalle, abordando el trabajo necesario y la necesidad del cliente.
Retraso en una tarea provoca el incumplimiento de las demás	Identificar tareas esenciales con mayor número de tareas dependientes y eliminar o aplazar aquellas que no sean necesarias.

## Conclusiones

Se llevo a cabo un proyecto de un software, para la administración de tareas, seguimiento de errores y gestión operativa de proyectos. La causa principal que motivo la realización del proyecto fue brindar una herramienta que ayude a los equipos a gestionar su trabajo, organizarse y mejorar el desempeño del equipo. Los objetivos planteados fueron alcanzados con éxito, ya que el sistema permite que sus usuarios colaboren entre ellos agrupando, clasificando y priorizando tareas que les permiten ser más ágiles y centrarse en lo importante.

Desde la perspectiva profesional la realización del proyecto me permitió emprender un recorrido completo en el proceso y desarrollo de una solución que dio como resultado un sistema, el cual requirió un gran esfuerzo en investigación, aspectos técnicos y codificación para lograr concluir con el proyecto, también requirió un gran trabajo de planificación y gestión de recursos y estimaciones de tiempos. Por ello me resulta satisfactorio haber cumplido con los objetivos planteados, que me permitieron adquirir conocimientos valiosos que como profesional lograre aplicar a lo largo de mi carrera profesional

## Demo

En el siguiente enlace se podrá visualizar el prototipo desarrollado para su ejecución

Link:

<https://xd.adobe.com/view/b27602ff-0a01-426c-ae79-ebb59e7e7f3d-e725/>



## Bibliografía

- Campos, G. E. (enero de 2018). *El 71% de las empresas en el mundo usa metodologías Agile*. Obtenido de cio mexico: <https://cio.com.mx/el-71-de-las-empresas-en-el-mundo-usa-metodologias-agile/>
- CPCIPC Consejo Profesional de Ciencias Informáticas de la Provincia de Córdoba. (s.f.). Obtenido de <https://www.cpcipc.org.ar/content/honorarios>
- Crisp. (2010). *s.f.*
- Crisp, (. (s.f.).
- Frías, E. R. (septiembre de 2018). *ira Software – Una aplicación para el seguimiento de errores web*. Obtenido de <https://es.atlassian.com/software/jira:https://estebanromero.com/herramientas-emprender-desarrollar-proyectos/jira-software-una-aplicacion-para-el-seguimiento-de-errores-web/>
- García, R. (9 de Julio de 2021). *Productividad y trabajo en equipo: Así puedes usar Trello*. Obtenido de <https://www.adslzone.net/reportajes/software/que-es-trello/>
- Gauchat. (2012).
- Herrera, M. (Marzo de 2009). *INGENIERÍA DEL SOFTWARE: METODOLOGÍAS Y CICLOS DE VIDA*. Obtenido de [https://www.academia.edu/9795641/INGENIER%C3%8DA\\_DEL\\_SOFTWARE\\_Y\\_METODOLOG%C3%8DAS\\_Y\\_CICLOS\\_DE\\_VIDA\\_Laboratorio\\_Nacional\\_de\\_Calidad\\_del\\_Software](https://www.academia.edu/9795641/INGENIER%C3%8DA_DEL_SOFTWARE_Y_METODOLOG%C3%8DAS_Y_CICLOS_DE_VIDA_Laboratorio_Nacional_de_Calidad_del_Software)
- <https://es.reactjs.org/>. (2019). Recuperado el Septiembre de 2021
- <https://nodejs.org/es/about/>. (2020).
- Ingeniería de Software*. (22 de Mayo de 2011). Obtenido de <http://gestionrrhusm.blogspot.com/2011/05/ingenieria-de-software-ingenieria-de.html>
- Mondayblog*. (3 de Junio de 2021). Obtenido de <https://monday.com/blog/es/gestion-de-proyectos/una-guia-completa-para-la-gestion-de-proyectos-en-2021/>
- Mora. (2002).
- Muradas, Y. (febrero de 2020). *Qué es Asana*. Obtenido de <https://asana.com/es:https://openwebinars.net/blog/que-es-asana/>
- Pecos. (2012).
- swhosting*. (s.f.). Obtenido de <https://www.swhosting.com/blog/transport-layer-security-tls-que-es-y-como-funciona/>
- Una guía completa para la gestión de proyectos en 2021*. (3 de Junio de 2021). Obtenido de mondayblog: <https://monday.com/blog/es/gestion-de-proyectos/una-guia-completa-para-la-gestion-de-proyectos-en-2021/>