

# Proyecto ComuniRed

## Ingeniería de Software

### Nombre y RUT de integrantes:

- Giovana Osorio Segovia 21432771-6
- Nicolás Gallo 21499359-7
- Sebastián Saa Ahumada 19839932-9

**Curso:** PTEC105 - NRC 10537

**Fecha:** 06-09-24

**Lapso:** 3er año de Ingeniería en Computación e Informática

### Control de versiones:

Versión	Descripción	Responsable	Fecha
1.0	Versión inicial del informe	Sebastián Saa A.	05/09/24
1.1	Definición del problema, objetivos, metodología y requerimientos.	Sebastián Saa A. y Giovana Osorio S.	07/09/24
1.2	Definición de la solución y metodología.	Sebastián Saa , Giovana Osorio y Nicolás Gallo.	09/09/24

## **Tabla de Contenidos**

Planteamiento del Problema.....	3
Planteamiento de la Solución.....	5
Justificación del Proyecto .....	8
Definición de los Objetivos del Proyecto.....	9
Metodología .....	10
Especificación de Requerimientos de Software .....	15
Recursos consultados .....	22

## Planteamiento del Problema

Nuestro proyecto, enfocado en la gestión de comunidades, está dirigido particularmente a condominios, villas, juntas de vecinos y todo tipo de comunidades residenciales, que busquen una herramienta para mejorar la organización y comunicación de la comunidad.

En un mundo más digitalizado, algunos de los desafíos a los que se enfrentan las comunidades de vecinos son el lograr una buena gestión y comunicación. La manera en que se abordan las diversas situaciones que ocurren suele ser informal y poco estructurada, lo cual puede generar problemas:

- **Estructura desorganizada:** Muchas veces no están claros los roles y responsabilidades que tiene cada integrante de la comunidad, lo cual trae problemas al momento de resolver problemas o en la gestión y transparencia de recursos.
- **Vía de comunicación:** La única vía de comunicación que presentan estas comunidades es generalmente un chat de WhatsApp, aplicación la cual actualmente no está diseñada para seguir temas importantes que se hayan discutido de manera grupal.
- **Baja participación:** Cuando no hay una estructura, medio de comunicación y organización, muchas veces los vecinos tienden a ignorar y/o desconectarse de asuntos comunitarios, lo que impacta directamente a la comunidad.
- **Gestión de finanzas:** Es difícil para algunas comunidades llevar una cuenta clara y transparente de las cuentas de esta, ya sea por deudores, malentendidos, actividades, etc.
- **Dependencia del administrador:** Algunos gestores de comunidades son gestionados por 1 solo administrador. Una sobrecarga de trabajo en este puede causar retrasos en actualizaciones, decisiones y transparencia.

La estructura y flexibilidad a través de una aplicación permitiría combatir estos problemas que los vecinos enfrentan a diario, donde todos podrían participar activamente en la comunidad, en un espacio claro y de fácil manipulación.

## Planteamiento de la Solución

Para este proyecto, se diseñará un software enfocado en la gestión de comunidades residenciales como condominios, villas o barrios. El software tiene como objetivo principal mejorar las comunidades en términos de organización, comunicación y transparencia. Esta aplicación web y móvil permitirá la gestión general de la comunidad, por ejemplo, gestión de finanzas, organización de eventos, reportes de actividades, manejo de incidente, gestión de usuarios y roles, ya sean miembros de la directiva, los conserjes o residentes, entre otras funcionalidades. Este software busca resolver problemas comunes, como la falta de organización, comunicación ineficaz, baja participación y gestión financiera inentendible y/o incompleta.

Se diferencia principalmente por su flexibilidad, ya que esta se adapta a cualquier comunidad, sin importar el tamaño o estructura, y también por su accesibilidad al poder llegar a comunidades que no contaban con una herramienta de este tipo. También, se busca la transparencia pues permitirá una mejor gestión de presupuestos, pagos de cuotas, reportes de actividades o incidentes y creación de eventos.

Las funciones generales que tendría el software son:

### 1. Gestión de usuarios y roles

- Los usuarios podrán ser directiva, conserjes, o miembros de la comunidad, cada uno teniendo permisos específicos que se ajusten a sus necesidades.
- Asignación, modificación y eliminación de roles y permisos por parte de la directiva, ya sean, asignar roles de propietario, residente, arrendatario o invitado.

### 2. Gestión de finanzas:

- Tendrá la función de registrar, consultar y pagar gastos comunes y multas a

través de diversos métodos de pagos.

- Transparencia en la gestión de presupuestos, mostrando un registro claro del historial de pagos y finanzas de la comunidad.

### 3. Comunicación y organización de eventos:

- Se proporcionará apartados para la planificación y coordinación de reuniones, permitiendo a los usuarios publicar avisos, convocar asambleas y coordinar eventos de manera efectiva.
- Se podrán enviar notificaciones con push, correos electrónicos y alertas dentro de la aplicación para mantener a todos los miembros informados y conectados, mejorando así la participación y la respuesta a actividades comunitarias

### 4. Reserva de espacios comunes:

- La plataforma incluirá un sistema de reservas para áreas comunes, como salones de eventos, canchas deportivas y estacionamientos, permitiendo a los usuarios ver la disponibilidad en tiempo real y hacer reservas de manera sencilla.
- Además, se podrán aplicar restricciones de uso, como límites de tiempo o la cantidad de reservas permitidas por usuario, y enviar confirmaciones de reservas a través de la app, asegurando una gestión ordenada y equitativa de los recursos.

### 5. Registro y gestión de incidentes:

- El software facilitará el registro y seguimiento de incidentes o problemas que surjan dentro de la comunidad, como fallos en servicios, quejas de los vecinos o situaciones de emergencia.

- Los conserjes podrán gestionar estas incidencias, actualizando su estado (pendiente, en progreso, resuelto) y comunicando el progreso a los miembros afectados, lo cual agiliza la resolución de problemas y mejora la satisfacción de los usuarios.

6. Control de acceso y seguridad:

- La aplicación permitirá registrar y monitorear la entrada y salida de visitantes, documentando datos como nombres, fechas, ubicación de destino, y motivo de la visita, ayudando a mantener un control riguroso sobre el acceso a la comunidad.
- Los conserjes tendrán acceso a módulos especiales que les facilitarán gestionar la seguridad y el mantenimiento, como la supervisión de entradas y salidas o el registro de actividades diarias.

## Justificación del Proyecto

La propuesta aborda directamente a las debilidades actuales al transformar un sistema informal y descentralizado en uno formal, organizado y accesible. Al permitir que los vecinos definan roles, se reduce la dependencia de un solo administrador, lo cual mejora la eficiencia y transparencia de la comunidad. Además, las funciones de la aplicación y el control de accesos provocan un entorno seguro para la comunidad.

La relevancia social de ComuniRed se basa en la capacidad para mejorar y facilitar la organización y colaboración en las comunidades residenciales. Esta sirve como herramienta que promueve la participación de los vecinos y fortalecer el vínculo de sus comunidades.

Esta aplicación está orientada a cualquier tipo de comunidad residencial, de esta manera se expande el impacto de estas plataformas en comunidades que nunca habían considerado utilizar una herramienta de gestión. Estas comunidades podrán tener una mejor gestión de sus intereses, preocupaciones y actividades a través de un espacio virtual con los canales necesarios y roles definidos.

Además, ComuniRed brinda a los integrantes del proyecto la oportunidad de desarrollar competencias claves como la gestión de proyecto ágiles, planificación de tareas y diseño de interfaz de usuario y de aplicaciones web y móvil.

Este trabajo mejora las habilidades técnicas, como la programación, la gestión de bases de datos, así como las habilidades blandas, el trabajo en equipo, la resolución de problemas y la comunicación efectiva con los stakeholders.



# Definición de los Objetivos del Proyecto

## Objetivo general:

Diseñar una aplicación para la gestión de comunidades de vecinos que permita la organización, asignación de roles y comunicación eficiente entre vecinos de condominios, villas, juntas de vecinos, barrios y edificios residenciales, lo cual fomenta la participación y actividad de la comunidad.

## Objetivos específicos:

1. Diseñar una plataforma atractiva e interactiva que permita a los usuarios crear espacios virtuales personalizados que se adapten al tamaño, características y necesidades de su comunidad, permitiendo a través de esta que definan roles, en un plazo de 4 meses.
2. Implementar funcionalidades de gestión de finanzas para impulsar y demostrar la transparencia de gastos comunes y facilitar el pago de cuotas, permitiendo que al menos un 80% de los usuarios adopten este sistema 3 meses después del lanzamiento.
3. Desarrollar un sistema de comunicación grupal que permita coordinar reuniones, publicar avisos y reportes, logrando que al menos 70% de los usuarios atiendan las notificaciones después de 2 meses de su implementación.
4. Evaluar el nivel de satisfacción de los usuarios durante los 6 primeros meses de uso, a través de encuestas de satisfacción, alcanzando una calificación de satisfacción de al menos 80% entre las comunidades que adopten la plataforma.

# Metodología

A continuación, se presenta una breve descripción y comparación de las metodologías RUP, Waterfall y Scrum, destacando sus características principales.

**Proceso Racional Unificado (RUP):** Es una metodología iterativa y orientada a casos de uso que divide el desarrollo en fases claramente definidas (Inicio, Elaboración, Construcción y Transición), donde se promueve una solución disciplinada, que consiste en organizar las tareas y responsabilidades de los miembros de un equipo. Este proceso suele implementarse en proyectos complejos y con equipos grandes, donde el riesgo debe gestionarse rigurosamente.

**Waterfall:** Es una metodología con un enfoque lineal y secuencial donde el progreso de un proyecto fluye constantemente hacia abajo a través de diferentes fases, visualmente es similar a una cascada. Una fase comienza cuando se ha completado la anterior, y la finalización de cada fase es terminal. Es ideal para proyectos cortos y poco cambiantes.

**Scrum:** Es una metodología ágil que organiza el trabajo en sprints cortos (1 a 4 semanas) donde se desarrollan entregas incrementales. Suele implementarse en entornos complejos, cuando los requisitos son cambiantes y se necesitan obtener resultados pronto. Además, define roles como el Product Owner, Scrum Master y equipo de desarrollo.

## Tabla Comparativa

Aspectos	RUP	Waterfall	Scrum
Flexibilidad	Moderada	Baja	Alta
Enfoque en los requisitos	Requisitos iniciales bien definidos, se pueden ajustar	Deben ser claros desde el inicio	Pueden evolucionar a medida que avanza el proyecto
Gestión de riesgos	Se mitigan desde	Los problemas	Pueden gestionarse

	fases tempranas	suelen descubrirse tarde	y mitigarse en cada sprint
<b>Documentación</b>	Alta	Alta	Mínima
<b>Planificación</b>	Alta	Alta	Moderada
<b>Entregas</b>	Incrementales y a largo plazo	El producto se entrega al final del ciclo	Incrementales y frecuentes
<b>Contacto con el cliente</b>	Se consulta en fases claves	Al inicio y al final	Comunicación constante, en cada sprint
<b>Escalabilidad</b>	Alta	Baja	Alta
<b>Ventajas</b>	<ul style="list-style-type: none"> <li>- Control sobre riesgos y fases</li> <li>- Entregas incrementales</li> <li>- Documentación detallada</li> </ul>	<ul style="list-style-type: none"> <li>- Claridad y estructura en proyectos predecibles</li> <li>- Fácil de entender y aplicar en entornos con pocos cambios</li> </ul>	<ul style="list-style-type: none"> <li>- Alta flexibilidad</li> <li>- Entregas rápidas</li> <li>- Feedback continuo y adaptación a cambios</li> </ul>
<b>Desventajas</b>	<ul style="list-style-type: none"> <li>- Complejo y costoso</li> <li>- Mucha documentación y formalismo</li> </ul>	<ul style="list-style-type: none"> <li>- No permite cambios</li> <li>- Los problemas pueden descubrirse tarde</li> </ul>	<ul style="list-style-type: none"> <li>- Requiere autoorganización y compromiso del equipo</li> <li>- Difícil de escalar sin adaptaciones previas</li> </ul>

Una vez revisada la comparación entre las metodologías RUP, Waterfall y Scrum, se puede concluir que Scrum es el marco de desarrollo ágil que mejor se adapta a las necesidades de este proyecto. Las razones por las cuales Scrum es la opción más adecuada son las siguientes:

- Scrum tiene definido sprints, los cuales permiten a los desarrollos mostrar un avance y recibir feedback del cliente al término de cada uno. Considerando que los clientes son vecinos, es probable que surjan nuevas necesidades y mejoras que dependen de la realidad de cada comunidad.
- También, se puede aprovechar el desarrollo por sprints para entregar versiones funcionales de la aplicación de manera incremental, como puede ser agregar la gestión de roles o control de pagos.
- Al descomponer el objetivo principal en actividades pequeñas, se podría entregar una versión temprana de este en un corto tiempo, para que los vecinos comiencen a usar la aplicación lo antes posible.

A continuación, se describen los pasos a seguir para implementar el marco de desarrollo ágil Scrum que permitirá adaptarnos a las necesidades que las comunidades nos presenten conforme avance el proyecto:

### **Paso 1: Creación del Product Backlog**

El Product Backlog es la lista de funcionalidades y tareas que se deben desarrollar a lo largo del proyecto. Para la aplicación de gestión de condominios, este backlog incluirá funcionalidades clave como la gestión de usuarios (directiva, conserjes y residentes), pagos de gastos comunes y multas, reservas de espacios comunes, y notificaciones y anuncios. Además, se podrán incorporar tareas técnicas como la integración de distintos métodos de pago, la gestión de bases de datos para el registro de usuarios y visitantes, y medidas de seguridad de la información, entre otros.

### **Paso 2: Planificación del Sprint**

En la reunión de planificación del sprint, el equipo seleccionará las funcionalidades más prioritarias del Product Backlog y las desglosa en tareas más pequeñas y manejables. Durante esta etapa, se asignan responsabilidades a cada miembro del equipo, quienes trabajarán en aspectos específicos del desarrollo de la aplicación. Por ejemplo, uno de los desarrolladores podría encargarse del backend del sistema de pagos, mientras que otro trabajaría en la interfaz de usuario para el módulo de reservas de áreas comunes.

### **Paso 3: Reuniones diarias**

A lo largo del sprint, se llevarán a cabo reuniones diarias cortas, de no más de 15-20 minutos. En estas reuniones, cada miembro del equipo informará sobre el trabajo realizado el día anterior, lo que planea hacer ese día y si enfrenta algún obstáculo que pueda afectar el desarrollo. Estas reuniones permiten una comunicación fluida y la resolución rápida de problemas.

#### **Paso 4: Desarrollo del Sprint**

Durante el sprint, el equipo se enfocará en desarrollar las funcionalidades acordadas durante la planificación. Cada sprint entrega un incremento funcional, es decir, una parte de la aplicación que ya es utilizable por los usuarios. Por ejemplo, al finalizar un sprint, el equipo podría haber desarrollado el sistema de pago de gastos comunes, permitiendo a los residentes comenzar a utilizar esa funcionalidad dentro de la aplicación.

#### **Paso 5: Revisión del Sprint**

Al finalizar cada sprint, se realizará una reunión de revisión en la que el equipo presentará el trabajo completado. Los stakeholders o interesados (como la directiva del condominio, la comunidad de residentes y otros usuarios) tendrán la oportunidad de probar las funcionalidades desarrolladas y ofrecer su retroalimentación. Este feedback es fundamental para ajustar y mejorar las futuras entregas.

#### **Paso 6: Retrospectiva del Sprint**

Una vez completada la revisión del sprint, se llevará a cabo una retrospectiva, donde el equipo evaluará el proceso de trabajo. En esta fase, se discutirá qué aspectos funcionaron bien y cuáles requieren mejora, lo que permitirá optimizar el rendimiento en los próximos sprints. Además, se podrán debatir ideas relacionadas con el diseño de la aplicación, la interfaz de usuario, y la usabilidad de las funcionalidades desarrolladas.

#### **Paso 7: Refinamiento del Product Backlog**

Con el avance de los sprints, el Product Backlog se ajustará de acuerdo con la retroalimentación obtenida de los usuarios (directiva, conserjes, comunidad) y las necesidades emergentes del proyecto. Esto permitirá priorizar nuevas funcionalidades o mejorar las ya existentes, garantizando que el producto final se adapte de manera óptima a los requerimientos del cliente.

# Especificación de Requerimientos de Software

**Descripción de los usuarios del sistema y las funcionalidades del software que éstos utilizarán.**

**Directiva:** Los miembros de la directiva se encargan de la gestión de la comunidad. Tienen acceso a amplias funciones de organización, creación y asignación de roles y de supervisión.

## Funcionalidades:

- Gestionar los gastos comunes de forma mensual.
- Activar, deshabilitar o modificar un usuario y sus roles.
- Gestionar solicitudes de mantenimiento y tareas.
- Crear anuncios para la comunidad.
- Consultar información sobre la comunidad, los espacios comunes, el pago de los gastos comunes y de las multas.

**Conserje:** El usuario conserje no está presente en todas las estructuras, se encarga principalmente de ejecutar y monitorear tareas relacionadas con la limpieza, mantenimiento y seguridad del condominio.

## Funcionalidades:

- Consultar información básica de la comunidad.
- Consultar y reservar espacios comunes a miembros de la comunidad.
- Ingresar y consultar un evento al libro de tareas y mantenimiento.
- Registrar y monitorear entrada y salida de visitas.
- Registrar correos e informar a la comunidad de la llegada de encargos.

**Comunidad:** Las funcionalidades de los miembros de la comunidad pueden ser restringidas por la asignación de roles de la directiva y se centran en el uso de los servicios que ofrece el sistema.

Funcionalidades:

- Consultar su información de gastos comunes.
- Ingresar observaciones y comentarios al cobro de sus gastos comunes.
- Consultar sus multas y morosidad.
- Consultar disponibilidad y reservar espacios comunes.
- Pagar gastos comunes y otros servicios.



### **Lista de Requerimientos de Usuario.**

1. La directiva debe poder agregar, eliminar y editar la información de los conserjes y de la comunidad.
2. La directiva debe poder generar y enviar facturas a los miembros de la comunidad.
3. La directiva debe poder registrar pagos y ver el historial de pagos.
4. La directiva debe poder gestionar solicitudes de incidencias y actividades.
5. La directiva debe poder enviar notificaciones y comunicados masivos a la comunidad.
6. La directiva debe poder supervisar finanzas (gastos comunes, multas, etc.) y la disponibilidad de espacios comunes realizadas por el conserje y la comunidad.
7. La directiva debe poder crear, modificar y eliminar documentos y reglamentos de la comunidad.
8. El conserje debe poder notificar a la comunidad del recibo de correos, facturas y paquetes.
9. El conserje debe poder registrar reservas e incidencias en espacios comunes.
10. El conserje debe poder visualizar información de los residentes.
11. El conserje debe poder registrar la entrada y salida de visitantes.
12. Los miembros de la comunidad deben poder visualizar sus facturas pendientes y pagarlas a través de la app.
13. Los miembros de la comunidad deben poder enviar comentarios u observaciones sobre el pago de los gastos comunes o multas.

14. Los miembros de la comunidad deben poder ver la disponibilidad de espacios comunes y realizar reservas.
15. Los miembros de la comunidad deben poder recibir notificaciones de la administración, directiva y conserje.

## **Lista de Requerimientos del Sistema:**

### Requerimientos funcionales:

- La aplicación debe permitir que los miembros de la directiva sean capaces de crear, asignar roles, modificar y eliminar usuarios para la comunidad y conserjes, controlando sus permisos y niveles de acceso en la aplicación.
- La aplicación debe ofrecer a la comunidad la posibilidad de pagar gastos comunes y multas a través de diferentes métodos de pago, como tarjeta de crédito, débito y transferencias bancarias.
- La aplicación debe permitir a la comunidad consultar el historial de pagos, descargar recibos y recibir notificaciones de pagos pendientes o vencidos.
- La aplicación debe permitir que el conserje pueda revisar la información básica de los residentes, como nombres, unidades y contacto.
- La aplicación debe permitir al conserje gestionar incidencias con la posibilidad de marcarlas como incompletas, en curso y completadas.
- La aplicación debe permitir al conserje registrar y gestionar el acceso de visitas, documentando nombre completo, RUT, fecha y hora, unidad de visita, patente (si aplica) y motivo de la visita.
- La aplicación debe permitir que tanto el conserje como la comunidad puedan reservar espacios comunes en tiempo real, visualizar la disponibilidad, modificar o cancelar reservas.
- La app debe enviar notificaciones de confirmación y gestionar las restricciones de uso de los espacios comunes, como límite de tiempo o cantidad de reservas permitidas.

- La app debe permitir que los miembros de la directiva generen anuncios dirigidos a la comunidad y/o conserjes, que se distribuirán mediante notificaciones push, correo electrónico o dentro de la app.
- La app debe permitir a la directiva programar la publicación de anuncios y recibir confirmaciones de lectura en casos de comunicaciones importantes.

Requerimientos no funcionales:

- La aplicación debe poder manejar hasta 1000 usuarios simultáneos sin afectar el rendimiento ni provocar tiempos de respuesta de más de 3 segundos.
- La aplicación debe implementar autenticación de dos factores para la directiva para mayor seguridad de acceso a la información.
- La aplicación debe estar disponible al menos el 98% del tiempo, con no más de 4 horas de inactividad por mantenimiento programado por mes.
- La aplicación debe ser compatible con los principales navegadores (Chrome, Safari, Firefox, Edge) y ejecutarse en dispositivos móviles iOS y Android sin problemas de rendimiento o usabilidad.
- El tiempo de carga inicial de la aplicación no debe exceder los 5 segundos para usuarios con una conexión a internet estándar.
- La aplicación debe optimizar el uso de CPU y memoria para que el consumo de recursos en dispositivos móviles no supere el 30% de la capacidad total.
- La aplicación debe implementar controles de acceso basados en roles para restringir el acceso a datos confidenciales y que solo los usuarios con los permisos adecuados puedan ver, modificar o eliminar información relevante.

- La aplicación debe tener un plan de recuperación ante desastres que asegure que los datos se puedan recuperar en 1 hora en caso de una falla o ataque y que se realice una copia de seguridad automática diariamente.
- El diseño de la interfaz de la app debe seguir principios de diseño centrado en el usuario para que cualquier usuario pueda completar una tarea en menos de 3 clics desde el menú principal.
- La aplicación debe permitir a los usuarios cambiar entre al menos 3 idiomas (español, inglés, portugués) para facilitar su uso en diferentes comunidades.

## Recursos consultados

Sommerville, I. (2011). Ingeniería de Software. 9na Ed. México D.F.: Pearson Educación

<https://www.comparasoftware.cl/administracion-de-condominios>

<https://www.comunidadfeliz.cl/>

Ortega, L. (2022, 22 octubre). Metodología RUP: ¿Qué es, cuál es su objetivo y cómo se utiliza? Lean Management Blog. <https://lean-management.site/rup/>

Metodologia RUP. (2017, 23 octubre). <https://metodolorup.blogspot.com/>

Ramírez, L. (2023, 2 febrero). Metodología Waterfall: Qué es y cuándo usarla. Thinking For Innovation. <https://www.iebschool.com/blog/metodologia-waterfall-que-es-y-cuando-usarla-management/>

Metodología en cascada: Ventajas e inconvenientes | SafetyCulture. (2024, 15 enero). SafetyCulture. <https://safetyculture.com/es/temas/metodologia-en-cascada/>

Los pros y los contras de la metodología de cascada. (2018, 10 agosto). <https://www.lucidchart.com/blog/es/pros-y-contras-de-la-metodologia-de-cascada>

Qué es SCRUM. (2021, 20 septiembre). Proyectos Ágiles. <https://proyectosagiles.org/que-es-scrum/>

Cómo funciona Scrum. (2021, 16 enero). Proyectos Ágiles. <https://proyectosagiles.org/como-funciona-scrum/>

Martins, Á. G. (2024, 17 junio). Ventajas y desventajas de usar el método SCRUM. Ofiprix. <https://www.ofiprix.com/es/blog/ventajas-desventajas-usar-metodo-scrum>

¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/scrum/>