

# **Digitalización y gestión de Notificaciones del Tribunal de Cuentas de Río Negro**

**Trabajo Final de Carrera  
Licenciatura en Sistemas**

**UNRN**

Universidad Nacional  
de **Río Negro**

**Alumno:** Nicolas Alfredo Dell

**Director:** Mauricio Tassara

<b>Introducción</b>	<b>6</b>
<b>Objetivos</b>	<b>7</b>
Generales	7
Específicos	7
<b>Contexto</b>	<b>8</b>
Problemática	8
Solución	9
Administración de notificaciones	9
<b>Características</b>	<b>11</b>
Base de datos centralizada	11
Interfaz de usuario intuitiva	11
Automatización de procesos	11
Acceso remoto	11
Respaldo de datos	11
Auditoría de usuarios	11
Mayor precisión	11
Mejora en la comunicación	11
<b>Digitalización</b>	<b>12</b>
Características	12
Precisión y consistencia	12
Almacenamiento eficiente	12
Facilidad de Distribución	12
Versatilidad	12
Almacenamiento a largo plazo	12
Reducción de costos	12
Facilita la automatización	13
Conversión de documentos en papel	13
Búsqueda de texto	13
Compartir y distribuir	13
Conservación de formato	13
Compatibilidad multiplataforma	13
<b>Marco teórico</b>	<b>14</b>
Metodología ágil	14
Principios del manifiesto ágil	14
<b>Scrum</b>	<b>14</b>
Reuniones	15
Planificación del backlog	15
Seguimiento del sprint	15
Revisión del sprint	15
Elementos del scrum	15
Product backlog	15
Sprint backlog	15
Incremento	15
Historia de usuario	16

Formato	16
Implementación	16
<b>Selección de herramientas</b>	<b>18</b>
Características	18
Eficiencia y productividad	18
Calidad del producto	18
Colaboración y comunicación	18
Adaptabilidad y escalabilidad	18
Reducción de costos y tiempo	18
Jira	19
Características	19
Gestión de proyectos	19
Colaboración mejorada	19
Seguimiento de problemas y tareas	19
Integración	19
Reporte y seguimiento	19
Metodologías ágiles	19
Escalabilidad	20
Facilita la gestión remota	20
Visual Studio Code	20
Código abierto	21
Buena experiencia de usuario	21
Angular	21
Laravel	21
Sintaxis expresiva y elegante	21
Migraciones de base de datos	21
Eloquent ORM	22
Rutas y controladores	22
Autenticación y autorización	22
TablePLUS	22
<b>Postman</b>	<b>22</b>
HTML	24
CSS	24
TypeScript	24
Tipado estático opcional	24
Mejora la productividad	25
Refactorización más segura	25
Mejora la legibilidad del código	25
Mejora la escalabilidad del código	25
MySQL	25
<b>Implementación</b>	<b>26</b>
Inicio de sesión	26
Cambio de pestaña / Navegación	26
Listado de externos	27

Alta de usuario externo	28
Modificación de usuario externo	28
Eliminación de usuario externo	29
Alta de grupo	29
Listado de grupos	29
Modificación de grupo	30
Eliminación de grupo	30
Asignación y designación de usuarios externos a grupos	31
Listado de plantillas	31
Alta de plantilla	32
Modificación de plantilla	32
Eliminación de plantilla	33
Envío de notificación	33
Envío de notificación a revisión	35
Listado notificaciones para revisión	38
Revisar y aprobar notificación	38
Listar notificaciones enviadas	39
Postman	40
Vista general	40
Variables de entorno	40
Métodos de usuarios externos	40
Jira	40
Sprint 0 - Product Backlog	40
Sprint 1	41
Sprint 1 - Kanban	41
<b>Conclusiones</b>	<b>42</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>43</b>
Anexo	44
Aclaración: Las palabras que se encuentran entre “{{“ y “}}” hacen referencia a las variables de entorno.	44
Obtener externos	44
Método	44
URL	44
Encabezados	45
Parámetros	45
Mensaje de respuesta con parámetros que pasen la validación	45
Datos obtenidos	45
Código de estado	45
Alta de usuario externo	46
Método	46
URL	46
Encabezados	46
Parámetros	46
Mensaje de respuesta con parámetros que pasen la validación	46

Datos obtenidos	47
Código de estado	47
Modificación de usuario externo	47
Método	47
URL	47
Encabezados	47
Parámetros	47
Mensaje de respuesta con parámetros que pasen la validación	48
Datos obtenidos	48
Código de estado	48
Eliminación de usuario externo	48
Método	48
URL	48
Encabezados	48
Parámetros	48
Mensaje de respuesta con parámetros que pasen la validación	49
Datos obtenidos	49
Código de estado	49
Mensaje de respuesta sin estar logueado en todos los endpoints	49
Datos obtenidos	49
Código de estado	49
<b>Referencias</b>	<b>50</b>

# Introducción

El propósito del presente informe es ofrecer un análisis sobre el desarrollo y la implementación de una aplicación web para el Tribunal de Cuentas de Río Negro, centrándose en los aspectos clave de su diseño, implementación técnica y potencial impacto. La aplicación web objeto de este informe se ha desarrollado con el objetivo de digitalizar los procesos de envío, recepción, respuesta y administración de notificaciones.

La organización del presente trabajo es el siguiente:

En un principio se presentarán los objetivos generales y específicos junto con el contexto del trabajo. También se abordará la problemática que surge debido a la falta de una herramienta que agilice los procesos y centre la información. Se adelantará de manera esquemática la solución planteada, haciendo hincapié en las características y beneficios de la digitalización y de la administración de notificaciones.

Luego, se presentará el marco teórico, con la definición y explicación de por qué se utilizaron metodologías ágiles para el desarrollo de la aplicación. Se detalla la forma de recopilar la información utilizada para establecer los requerimientos del sistema.

A continuación, se explicarán y presentarán en detalle las herramientas que fueron utilizadas para realizar la implementación del sistema. Se explicará por qué es importante seleccionar herramientas antes de desarrollar y cuáles son los beneficios de cada una de las elegidas.

Finalmente, se mostrarán imágenes de la aplicación junto con los pasos que deberían realizar los agentes para utilizar el sistema. Se mostrarán imágenes del proyecto en Postman, con la estructura de carpetas utilizadas y ejemplos de endpoints desarrollados. Y, por último, imágenes de la aplicación Jira mostrando el Sprint 0 junto al product backlog y el Sprint 1 junto a las historias de usuarios.

# Objetivos

## Generales

- Implementar un sistema para digitalizar los procesos de envío, recepción y respuesta de notificaciones.
- Desarrollar un sistema de administración de notificaciones que posea una interfaz de usuario amigable y con usabilidad simple.

## Específicos

- Mejorar los procesos de armado, envío y seguimiento de notificaciones para tener una mayor eficiencia y disminuir los tiempos de comunicación.
- Centralizar y organizar la información obtenida a partir de las notificaciones enviadas para mejorar su acceso y usabilidad.

# Contexto

## Problemática

En el Tribunal de Cuentas de Río Negro, los procesos para informar a los externos<sup>1</sup> por parte de los agentes<sup>2</sup> no están definidos. Los primeros son quienes deben notificar a los segundos de la manera más rápida y eficaz posible sobre las novedades de cada caso particular.

Al no tener un sistema único para llevar adelante estos procesos, los agentes seleccionaron los medios que consideraron adecuados en su momento para realizar la notificación. Al no haber ningún canal preestablecido, se procedió a la utilización de diversos medios para informar a los externos: WhatsApp, Telegram, Gmail, Hotmail, etc. Al utilizar varias plataformas, sin un sistema de comunicación unificado, los datos se dispersan y llegan a ser difíciles de encontrar, con lo que no se puede organizar la información y acceder a ella fácilmente.

Un ejemplo común de una situación problemática hipotética es el siguiente: si un agente dejara de trabajar en el Tribunal o se encontrara en receso ¿cómo podrían los demás agentes acceder a la información que está en el correo personal o whatsapp del agente? O peor aún ¿qué pasaría si se eliminaran los mensajes de Whatsapp o los correos de ese agente? La información se perdería permanentemente, sin dejar registro alguno. Y si un agente dejara de prestar servicios en el Tribunal, ¿se llevaría la información institucional en sus cuentas personales?

La falta de un sistema dedicado para que los agentes informen a los externos sobre los proyectos genera pérdidas de tiempo y aumento en la cantidad de pasos en el proceso, lo que lo hace ineficiente. Esto ocurre tanto desde el lado del agente como desde el lado del externo.

Por ejemplo, una vez que los externos son informados sobre determinado proyecto, deben ir personalmente al Tribunal con la documentación necesaria. Dado que no hay comunicación previa, se generan muchas veces situaciones en las que hay que volver a realizar la presentación porque dicha documentación puede no ser aceptable por errores o bien no estar completa.

En cuanto a los agentes, una notificación se considera en condiciones de ser enviada a un externo cuando tiene la información pertinente y la plantilla adecuada. En ese caso es enviada primero por algún canal de comunicación a un superior para su posterior autorización. Una vez completado este paso, se envía al externo. El rastreo de una notificación enviada a un externo es complicado:

- Si el envío se hizo por correo electrónico, debe buscarse por destinatario y por fecha, dado que puede ocurrir que se hayan enviado varios correos al mismo externo
- Para buscar por destinatario es necesario recordar el nombre y apellido del externo
- Es necesario revisar también los documentos adjuntos enviados y las posibles respuestas en caso de haberlas
- Si el envío se hizo por Whatsapp es necesario buscar por el nombre agendado, que puede no coincidir exactamente con el del externo

---

<sup>1</sup> Ente o individuo que no forma parte del Tribunal de Cuentas de Río Negro.

<sup>2</sup> Individuo que presta servicios al Tribunal de Cuentas de Río Negro.

- Se deben buscar no solo los mensajes originales sino también los referidos, que tengan relación con la notificación enviada

## Solución

Frente a la problemática planteada anteriormente, que afecta a todos los agentes del Tribunal de Cuentas de Río Negro y a los externos, se plantea como solución la creación de una aplicación web denominada “Chasqui”.

En esta aplicación, los agentes crean notificaciones en el sistema. Por cada notificación creada se envía un correo electrónico a los destinatarios en el que se les informa cuáles son los documentos que se encuentran involucrados. Esos documentos se envían como adjuntos en formato PDF. También encontrarán un enlace que los redirigirá al sistema para indicar que recibieron dicha notificación. Desde el mismo sistema podrán adjuntar la documentación solicitada por los agentes.

Todos los agentes poseen una computadora individual y acceso a Internet dentro del establecimiento por lo que pueden utilizar el sistema web sin ningún inconveniente. Además, en los casos en que, por alguna circunstancia, deban trabajar de forma remota podrán continuar su labor desde cualquier lugar.

Por último, otra ventaja es que se realizan *backups* periódicamente para que, en caso de algún problema, se asegure la conservación de la información hasta el día anterior y no se pierda permanentemente.

Esta solución informática tiene como objetivos principales:

- la administración eficiente de notificaciones por parte de los agentes, mediante una interfaz de usuario amigable y de usabilidad simple
- una mejora en los procesos de armado, envío y seguimiento de notificaciones para tener una mayor eficiencia y disminuir los tiempos de comunicación
- la centralización y organización de la información obtenida a partir de las notificaciones enviadas para mejorar su acceso y usabilidad
- la digitalización (despapelización) de la documentación generada en los procesos de envío, recepción y respuesta de notificaciones

## Administración de notificaciones

El objetivo del sistema es gestionar eficientemente las notificaciones junto con sus correos, de manera que la información de cada notificación pueda ser seguida y vista en forma detallada.

Cuando los agentes ingresan al sistema lo hacen además a grupos de trabajo ya asignados, y al encontrarse dentro de éstos, las notificaciones pueden ser vistas por todos los agentes del mismo grupo. Todos estos datos son a su vez verificados por un supervisor, quien además se asegura que esté todo listo para ser enviado.

Otra ventaja es la rápida verificación de los *mails* enviados para tener la certeza de si fueron recibidos por el externo. De no haber sido recibido el mail, se comprobará la dirección de correo electrónico, que estará guardada en el sistema y archivada por asunto o receptor. Así también se pueden visualizar las respuestas de los externos en archivos PDFs, con los archivos que se solicitaron. De estar correctos serán aceptados, o en caso contrario, rechazados con la posterior notificación al externo sobre la falla.

Se tiene una lista de las plantillas creadas que sirven como base para enviar correos. Los usuarios deben poner entre arrobas las palabras que desean modificar en el correo para luego realizar el envío del mismo.

Estas listas de notificaciones enviadas se pueden utilizar como historial y acceder rápidamente a ellos con toda la información necesaria para saber qué es lo que pasó con ellos. Además tienen filtros que sirven para hacer búsquedas específicas. Estas notificaciones vienen devueltos de manera descendente por fecha de creación.

Administrar las notificaciones permite realizar los pasos más eficientemente y reducir los tiempos de trabajo.

# Características

## Base de datos centralizada

Cuenta con una base de datos centralizada que almacena todos los datos relevantes. Esto permite acceder a información actualizada de manera más eficiente por parte de los agentes.

## Interfaz de usuario intuitiva

El sistema cuenta con una interfaz de usuario amigable y fácil de usar que permite a los agentes navegar y realizar tareas sin dificultad.

## Automatización de procesos

Este sistema permite automatizar procesos repetitivos y rutinarios, lo que ahorra tiempo y reduce errores por parte de los agentes, como el alta un externo o la creación de plantillas.

## Acceso remoto

El sistema es una aplicación web por lo que permite el acceso remoto y facilita la gestión desde diferentes ubicaciones por parte de los agentes.

## Respaldo de datos

El sistema cuenta con mecanismos de respaldo de datos para prevenir la pérdida de información crítica.

## Auditoría de usuarios

La aplicación tiene la capacidad de llevar a cabo un seguimiento de las actividades realizadas por los usuarios.

## Mayor precisión

La automatización del sistema reduce la posibilidad de errores por parte de los agentes, lo que mejora la precisión de los datos y la toma de decisiones.

## Mejora en la comunicación

Facilita la comunicación y colaboración entre diferentes áreas al proporcionar acceso a la misma fuente de información.

# Digitalización

Se busca digitalizar la documentación necesaria para informar a los externos sobre la documentación que debe ser presentada al Tribunal. Así, se la convierte del papel al formato digital, para representar datos e imágenes en formato electrónico.

La ventaja principal de la digitalización es que los datos digitales pueden ser procesados y manipulados fácilmente mediante software. Se puede editar texto, realizar búsquedas y hasta análisis.

Los datos digitales se pueden almacenar en dispositivos de almacenamiento electrónicos y se pueden transmitir a través de redes de comunicación.

La digitalización permite una mayor eficiencia en el manejo de la información y la posibilidad de conservar y compartir datos de manera más sencilla que con formatos analógicos.

Además, por otro lado, la digitalización permite también convertir estos datos digitales en analógicos, por lo que se puede convertir en papel el archivo digital.

## Características

### Precisión y consistencia

La digitalización permite tener una representación precisa y consistente de datos e imágenes. Los datos digitales no sufren degradación con el tiempo y se pueden replicar sin pérdida de calidad.

### Almacenamiento eficiente

Los datos digitales ocupan menos espacio físico en comparación con medios analógicos como el papel. Además, se pueden almacenar en cualquier dispositivo de almacenamiento electrónico.

### Facilidad de Distribución

Los datos digitales se pueden distribuir de manera rápida y sencilla a través de redes de comunicación, como Internet o redes locales. Esto facilita la colaboración y el intercambio de información.

### Versatilidad

Los datos digitales se pueden convertir y utilizar en una variedad de formatos y plataformas. Por ejemplo, en este caso los documentos se almacenarán en archivos con extensión PDF.

### Almacenamiento a largo plazo

Los datos digitales se pueden almacenar de manera segura a largo plazo, y es posible realizar copias de seguridad y replicación para evitar pérdidas.

### Reducción de costos

La digitalización puede reducir los costos asociados con la impresión, el almacenamiento físico y la distribución de información. Además, disminuye la necesidad de material impreso.

## Facilita la automatización

Los datos digitales son ideales para la automatización de tareas, ya que las computadoras pueden procesarlos y tomar decisiones basadas en algoritmos.

## Conversión de documentos en papel

Se pueden digitalizar documentos impresos, como contratos, informes, facturas, formularios, libros y otros documentos en papel, convirtiéndolos en archivos PDF. Esto facilita el almacenamiento y la gestión electrónica de documentos.

## Búsqueda de texto

Dentro de los archivos PDF se pueden realizar búsquedas de texto. Esto significa que se pueden encontrar fácilmente palabras clave dentro de un PDF, lo que agiliza la localización de información específica.

## Compartir y distribuir

Los archivos PDF son ampliamente compatibles y se pueden compartir fácilmente por correo electrónico o a través de servicios en la nube. La mayoría de las personas pueden abrir y ver archivos PDF sin necesidad de software especializado.

## Conservación de formato

Los archivos PDF preservan el formato original del documento, incluidas las fuentes, imágenes y diseño. Esto es útil para mantener la apariencia del documento tal como estaba en su formato original.

## Compatibilidad multiplataforma

Los archivos PDF se pueden abrir en una amplia variedad de dispositivos y sistemas operativos, incluidos ordenadores, tabletas y teléfonos móviles.

# Marco teórico

## Metodología ágil

Para llevar adelante el sistema se seleccionaron las metodologías ágiles. Estas permiten crear software rápidamente sin dejar de lado la calidad, permiten realizar cambios constantes y reducir los tiempos de desarrollo. Es fundamental en entornos donde hay incertidumbres. Se realizan entregas pequeñas, permitiendo al cliente valorar el producto al ver entregas continuas. Los ciclos cortos de entrega nos ayudan a disminuir los riesgos.

Estas metodologías se encuentran centradas en iteraciones y en la comunicación. Las iteraciones son realizadas en plazos de tiempos cortos.

Se tiene más en cuenta el software que funciona por encima de la documentación. La funcionalidad se ha completado cuando sus características pasan todas las pruebas y pueden ser utilizados por un usuario final.

En estas metodologías no es necesario conocer el producto desde un principio, el desarrollo puede comenzar una vez teniendo los objetivos generales, sin tantos detalles de los requerimientos. Por lo que, si los requisitos cambian, las modificaciones y las correcciones de errores no son tan costosas. Se trabaja en equipo entre el cliente y los desarrolladores con comunicación diaria para evitar errores y documentación innecesaria.

## Principios del manifiesto ágil

- Satisfacer al cliente con entregas tempranas y continuas de software.
- Aceptar que los requerimientos cambien.
- Entregar software funcional frecuentemente con un periodo de tiempo más corto posible.
- Responsables del negocio y desarrolladores trabajan juntos de forma diaria.
- El software funcional es la medida principal de progreso.
- Atención continua a la excelencia técnica y al buen diseño para mejorar la agilidad.
- En intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

## Scrum

Scrum fue seleccionado, frente al resto de metodologías ágiles, porque permite crear productos en menos tiempo, menos costos y de buena calidad. Permite agilidad, flexibilidad y trabajar frente a incertidumbres.

Esta metodología ágil tiene como base la creación de ciclos breves (sprints) para el desarrollo. Los sprints se gestionan a través de reuniones diarias (dailies).

Tiene cinco fases que definen el ciclo de desarrollo ágil:

- Concepto: Definir generalmente las características del producto y asignar el equipo que se encargará de su desarrollo.
- Especificación: Se establecen los límites del desarrollo del producto, como los costes y agendas. Se construye el producto a partir de las ideas principales y se comprueban las partes realizadas y su impacto en el entorno. Esta fase se repite en cada sprint y consiste en desarrollar y revisar los requisitos generales, mantener la

lista de las funcionalidades que se esperan y plan de entrega con fechas de las versiones, hitos e iteraciones.

- Exploración: Incremento del producto, se añaden las funcionalidades de la fase de especulación.
- Revisión: Se revisa lo que se ha construido y se contrasta con el objetivo deseado.
- Cierre: Se acuerda en la fecha acordada una versión del producto final. El cierre no indica que se finalizó, sino que seguirán habiendo cambios denominados "mantenimiento".

## Reuniones

### Planificación del backlog

Se define un documento en donde se reflejan los requisitos del sistema por prioridades. Se define la planificación del Sprint 0, donde se establecen los objetivos y trabajo para esta iteración. Se obtiene de esta reunión un sprint backlog que es la lista de tareas.

### Seguimiento del sprint

Reuniones diarias donde se tratan 3 temas. Que se realizó desde la reunión anterior, que se realizará hasta la próxima reunión y que inconvenientes surgieron y cómo solucionarlo para continuar.

### Revisión del sprint

Se realiza cuando finaliza el sprint la revisión del incremento que se ha generado. Se presentan los resultados finales y una demo/versión, permite mejorar el feedback con el cliente.

## Elementos del scrum

### Product backlog

Es donde se almacenan todas las necesidades y los requisitos que irá adquiriendo el producto a medida que avanzan los sprints. Cada objetivo suele ser expresado en historias de usuarios. Es necesario determinar los objetivos principales que pueden ir variando en cada iteración y se pueden generar nuevos requisitos.

### Sprint backlog

Es la lista de tareas que se realizan en un sprint. Las tareas se asignan a una persona y se indica el tiempo que tarda en terminarlas.

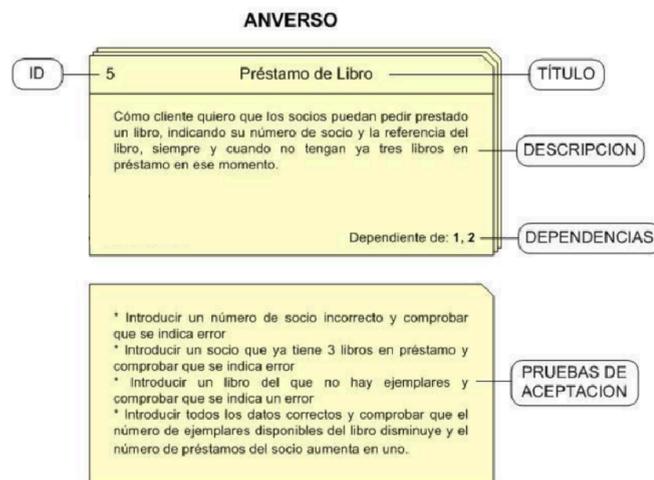
### Incremento

Parte desarrollada de un sprint, es una parte terminada y totalmente operativa.

## Historia de usuario

Es uno de los elementos de Scrum. En ellas se redactan las descripciones de las funcionalidades que va a tener el software.

### Formato



**ID:** identificador de la historia de usuario.

**Título:** título descriptivo de la historia de usuario.

**Descripción:** descripción breve de la historia de usuario.

**Dependencias:** elementos, tareas o condiciones que deben cumplirse antes de que la historia pueda ser implementada.

**Pruebas de aceptación:** condiciones que deben cumplirse para que una historia sea considerada completa y cumpla con los requisitos del usuario final o cliente.

## Implementación

En base a la primera reunión con los agentes se planificó el Sprint 0, definimos los objetivos generales del sistema y creamos el backlog inicial con las principales funcionalidades a desarrollar. Durante la reunión, el equipo de desarrollo y los agentes discutimos las prioridades, y como resultado, se generó un Sprint Backlog con tareas asignadas a cada miembro del equipo.

Cada sprint tuvo una duración de dos a tres semanas, y se incluyeron funcionalidades clave como:

Sprint 1: Proceso de creación de externos.

Sprint 2: Proceso de creación de plantillas.

Sprint 3: Implementación del módulo de creación de grupos junto con la asignación de externos a los mismos.

Sprint 4: Implementación del módulo de creación de notificaciones, junto con la visualización del historial y el filtrado por diferentes criterios.

Por otro lado, las reuniones diarias fueron clave para asegurar que todos los miembros del equipo estuvieran alineados y conscientes del progreso del proyecto. Estas reuniones se realizaban cada mañana y tenían una duración máxima de 15 minutos. Cada miembro del

equipo reportaba los avances desde la última reunión, mencionaba las tareas que iban a trabajar en el día y destacaba cualquier inconveniente que se había presentado, los cuales eran discutidos para buscar una solución rápidamente.

La Revisión del Sprint se llevaba a cabo luego de finalizar cada sprint, y su propósito era presentar a los agentes los resultados obtenidos durante ese Sprint. Las revisiones se centraban en la inspección del incremento desarrollado, la parte del sistema que fue completada y que ya estaba en funcionamiento. Los agentes participaban activamente en cada revisión, brindando retroalimentación sobre el módulo presentado. Se obtuvieron comentarios que permitieron ajustar el desarrollo en los siguientes sprints. El equipo evaluaba si se habían cumplido todos los objetivos del sprint. En los casos donde alguna tarea no hubiera sido finalizada, se analizaba la razón, ya fuera por falta de tiempo, problemas técnicos, o cambios en los requisitos.

A partir de los comentarios y la evaluación, se ajustaba el Product Backlog. Se añadían nuevas historias de usuario, se reordenaban prioridades, y se ajustaban los requisitos. Durante la revisión se realizaba una demostración funcional del incremento para que los agentes pudieran interactuar con el sistema.

# Selección de herramientas

La selección de herramientas para el desarrollo de software tiene un impacto significativo en la eficiencia, calidad, comunicación, costos y adaptabilidad.

## Características

### Eficiencia y productividad

Seleccionar las herramientas adecuadas aumenta la eficiencia del desarrollo al automatizar tareas, optimizar procesos y reducir el tiempo necesario para realizar actividades.

### Calidad del producto

Existen herramientas especializadas que ayudan a realizar pruebas, depuración y análisis que garantizan la calidad del software producido, lo que reduce errores y problemas en el producto final.

### Colaboración y comunicación

En este punto se incluyen herramientas de gestión de proyectos, sistemas de seguimiento, plataformas de colaboración, y sistemas de control de versiones facilitan la comunicación y colaboración entre equipos.

### Adaptabilidad y escalabilidad

Las herramientas adecuadas deben ser adaptables y escalables para satisfacer las necesidades cambiantes a medida que el proyecto crece.

### Reducción de costos y tiempo

La elección adecuada de herramientas puede reducir los costos y el tiempo de desarrollo. Herramientas de código abierto, por ejemplo, pueden ser gratuitas, reduciendo los gastos, mientras que las herramientas de automatización pueden acelerar el proceso de desarrollo.

# Entorno de desarrollo

## Jira

Jira es una plataforma popular de software desarrollada por Atlassian que se utiliza para la gestión de proyectos y el seguimiento de problemas. Está diseñada principalmente para equipos de desarrollo de software. Proporciona herramientas para la planificación, seguimiento y gestión de proyectos ágiles y tradicionales.

Jira se utilizó para plasmar los requerimientos del sistema que se obtuvieron en la etapa de recopilación de información. A lo largo de los sprints se fueron asignando los requerimientos que debían completarse en cada uno. Esto permitió tener un mayor control sobre cómo avanzaba el sistema. Se pudo verificar que requerimientos se terminaron y cuales presentaron algún inconveniente para continuar. Se utilizó el product backlog y el Kanban principalmente.

## Características

### Gestión de proyectos

Permitió la creación y gestión del proyecto, lo que facilitó la planificación, el seguimiento y la colaboración en el proyecto. Jira permitió planificar, organizar y dar seguimiento al proyecto de manera efectiva, lo que contribuyó a un mayor productividad y cumplimiento de plazos.

### Colaboración mejorada

Jira brindó la posibilidad de comentar, adjuntar archivos y mantener un registro de la comunicación relacionada con los problemas y tareas.

### Seguimiento de problemas y tareas

Se pudo registrar, asignar y dar seguimiento a problemas, tareas y mejoras, esto ayudó a evitar pérdida de información y a mantener un registro claro de las actividades.

### Integración

Jira se integra con una variedad de otras herramientas y servicios, como GitHub, Confluence, Bitbucket y muchas más, lo que facilita la colaboración y el flujo de trabajo.

### Reporte y seguimiento

Jira ofrece generación de informes y paneles de control para que los equipos puedan supervisar el progreso y el rendimiento del proyecto.

### Metodologías ágiles

Jira es compatible con metodologías ágiles como Scrum y Kanban, lo que facilitó la planificación y el seguimiento del proyecto.

## Escalabilidad

Jira es escalable y puede adaptarse a las necesidades de equipos pequeños y grandes, lo que la hace adecuada para una variedad de organizaciones.

## Facilita la gestión remota

Con la capacidad de acceder a Jira en línea, los equipos pueden gestionar proyectos y tareas desde cualquier ubicación, lo que es especialmente beneficioso para el trabajo remoto y la colaboración en equipos distribuidos.

En resumen, Jira es una herramienta versátil que se utiliza para gestionar proyectos, realizar un seguimiento de problemas y fomentar la colaboración en equipos de desarrollo de software. Ayuda a los equipos a ser más eficientes y a mantener un registro claro de su trabajo y sus tareas pendientes.

# Visual Studio Code

Esta herramienta es un editor de código fuente desarrollado por Microsoft. Es una herramienta actualmente popular y ampliamente utilizada por desarrolladores de software de todo el mundo. Es conocido por su simplicidad y eficiencia a la hora de programar. Es potente y es multiplataforma, puede ser utilizado en cualquier ordenador que posea cualquier sistema operativo actual (Windows, Linux y macOS).

Posee Intellisense, este ofrece características avanzadas de edición de texto como el resaltado de sintaxis, el autocompletado, refactorización, búsqueda y reemplazo. que proporciona terminaciones inteligentes basadas en tipos de variables, definiciones de funciones y módulos importados.

Tiene depuración de código integrado. Proporciona herramientas de depuración integradas que permiten a los desarrolladores depurar sus aplicaciones directamente desde el editor.

Además cuenta con comando GIT incorporados. Permite trabajar con GIT desde un principio y otros proveedores SCM (sistemas de control de versiones) fácilmente. Solucionar diferencias, visualizar los cambios entre versiones, pushear directamente desde el editor. Lo que facilita el seguimiento de cambios y colaboración en proyectos.

La terminal integrada que permite a los desarrolladores ejecutar comandos y scripts directamente desde el editor.

Por otra parte, soporta múltiples terminales y paneles, lo que facilita la organización de proyectos pudiendo abrir varias terminales para ejecutar diferentes comandos o abrir varios archivos para visualizarlos.

Permite instalar extensiones para agregar más características, ya sea para cambiar temas, depuradores, conectarse a servicios, agregar comandos, es decir para agregar funcionalidades adicionales. Es totalmente personalizable su interfaz de usuario para que cada desarrollador de software lo personalice como lo desea.

Finalmente, permite programar cualquier lenguaje de programación.

## Código abierto

Se trata de una herramienta de código abierto, por lo que el código fuente puede ser accedido por cualquiera, está “abierto” al público. Esto hace que cualquiera pueda aportar mejoras y/o soluciones a problemas del sistema.

## Buena experiencia de usuario

Al ser casi en su totalidad personalizable los usuarios pueden adaptar su marco de trabajo de acuerdo a sus gustos y preferencias.

## Angular

Es un marco de trabajo (framework) para el desarrollo de software web creado por Google. Es utilizado para crear aplicaciones web dinámicas y de una sola página (SPA, single page application) lo que significa que su contenido se carga una vez y las páginas siguientes se actualizan de forma dinámica sin necesidad de recargar la página completamente.

Funciona en cualquier tipo de proyecto, sin importar su escala. Desde proyectos pequeños hasta grandes.

Se basa en el lenguaje de programación TypeScript que proporciona una amplia cantidad de herramientas y funcionalidades que facilitan la creación de aplicaciones.

Se encuentra basado en componentes reutilizables. Los componentes son unidades independientes de código que se encargan de partes específicas de la interfaz de usuario. Estos componentes se comunican entre ellos para construir así la aplicación en conjunto. Angular ofrece enlaces de datos bidireccionales, lo que significa que los cambios en el modelo de la aplicación se reflejan automáticamente en la vista y viceversa.

## Laravel

Es un popular framework de desarrollo web que está escrito en PHP. Sigue el patrón de arquitectura MVC (Modelo-Vista-Controlador), lo que proporciona una estructura organizada y limpia para el desarrollo de aplicaciones web.

## Sintaxis expresiva y elegante

Se destaca por su sintaxis clara y expresiva, lo que hace que el código sea fácil de leer y escribir.

## Migraciones de base de datos

Ofrece un sistema de migraciones para versionar y controlar los cambios en la estructura de la base de datos, lo que facilita la gestión de bases de datos.

## Eloquent ORM

Proporciona un ORM (Object-Relational Mapping) llamado Eloquent que simplifica la interacción con la base de datos usando objetos y facilita la gestión de datos.

## Rutas y controladores

Facilita la definición de rutas y la creación de controladores para manejar las solicitudes HTTP de una manera clara y organizada.

## Autenticación y autorización

Incluye un sistema de autenticación que se puede implementar rápidamente, así como opciones para el control de acceso y la autorización.

## TablePLUS

Es una herramienta popular de administración de base de datos multiplataforma que ofrece una interfaz gráfica para gestionar y trabajar con diferentes tipos de datos. Es compatible con una amplia variedad de sistemas de gestión de bases de datos como MySQL, PostgreSQL, SQLite, etc.

Tiene una interfaz fácil de usar y personalizable que permite a los usuarios administrar bases de datos sin la necesidad de escribir comandos SQL manualmente.

Permite editar datos directamente en la tabla con funciones como búsqueda, ordenación y filtrado. También ofrece una vista de árbol para explorar la estructura de la base de datos.

Ofrece características de colaboración que permiten a varios usuarios trabajar simultáneamente en la misma base de datos, además de proporcionar atajos de teclado personalizables para aumentar la productividad.

Es compatible con una variedad de sistemas de gestión de bases de datos, lo que lo hace útil para aquellos que trabajan con diferentes tipos de bases de datos.

## Postman

Es una plataforma de colaboración y desarrollo de API que proporciona herramientas para que los desarrolladores puedan diseñar, probar y depurar APIs de manera más eficiente.

Permite crear y enviar solicitudes HTTP, como GET, POST, PUT, DELETE, etc. para probar APIs y servicios web. Proporciona un entorno intuitivo para ingresar parámetros, encabezados, datos de formulario, entre otros.

Permite organizar y guardar solicitudes en colecciones, facilitando la gestión y la reutilización de las solicitudes para pruebas o desarrollo posterior.

Facilita la automatización de pruebas y la creación de scripts para pruebas repetitivas de APIs, permitiendo verificar el funcionamiento correcto de las APIs en diferentes escenarios.

Permite crear documentación de APIs, lo que ayuda a compartir información sobre el uso y la implementación de APIs con otros desarrolladores o equipos.

Ofrece características para colaborar en equipos, como compartir colecciones, ambientes y pruebas, lo que facilita la cooperación entre desarrolladores, testers y equipos de operaciones.

Proporciona herramientas para supervisar el rendimiento de las API, permitiendo identificar problemas o cuellos de botella en el rendimiento de los servicios.

# Tecnologías aplicadas

## HTML

Es un lenguaje de marcas de hipertexto. Es el lenguaje estándar utilizado para crear y diseñar páginas web. HTML permite definir el significado y la estructura del contenido web mediante una serie de elementos, etiquetas y atributos. Estos elementos permiten organizar el contenido de la página, como texto, imágenes, videos, enlaces, formularios, entre otros. Cada etiqueta se diferencia por su nombre, el nombre de cada etiqueta está rodeado de un “<” y “>”. Además, cada etiqueta define diferentes tipos de contenido y tiene una función específica.

## CSS

Las “hojas de estilos en cascada” son un lenguaje de estilos. CSS permite definir la apariencia/aspecto de las páginas web, haciéndolas más atractivas para el usuario mejorando la experiencia de usuario. Se utiliza para definir el diseño, los colores, las fuentes, los espacios en blanco, el tamaño de los elementos, entre otros. Cada desarrollador especifica los estilos que se le aplican a uno o varios elementos de la página web.

Mediante regla de estilo, el cual consiste en selectores y declaraciones, CSS permite controlar el diseño y la presentación de múltiples páginas web a la vez. Los selectores que se utilizan en CSS apuntan a los elementos específicos del HTML y las declaraciones definen cómo se verán.

## TypeScript

Este lenguaje de programación desarrollado por Microsoft y basado en JavaScript permite definir el comportamiento/funcionamiento de las páginas web. Cualquier código válido de JavaScript es código de TypeScript válido. TypeScript es una extensión de JavaScript que proporciona elementos adicionales para el desarrollo de aplicaciones robustas.

Posee características de tipado estático opcional a las variables, lo que permite detectar errores y prevenir errores durante el desarrollo. El sistema de tipos puede identificar posibles problemas en el código antes de su ejecución. Esto, además, da robustez al código y legibilidad.

Por otro lado, ofrece otras características como la utilización de clases, interfaces, módulos y otros conceptos de la programación orientada a objetos. Esto facilita la organización y la estructuración del código en aplicaciones más grandes.

## Tipado estático opcional

Permite definir tipos para las variables, los parámetros de funciones, etc, lo que ayuda a detectar errores durante el desarrollo. Esto permite hacer a un código más robusto al prevenir errores comunes antes de su ejecución.

## Mejora la productividad

El sistema de tipos ofrece información detallada sobre el código, lo que facilita la comprensión y el mantenimiento. Además, los IDEs pueden ofrecer funcionalidades avanzadas, como autocompletado de código, gracias a esta información tipada.

## Refactorización más segura

Los cambios en el código pueden ser realizados con más confianza debido a la detección de errores en el tiempo de compilación. Esto permite reestructurar el código de manera más segura y eficiente.

## Mejora la legibilidad del código

Los tipos y la capacidad de definir interfaces y estructuras claras pueden hacer que el código sea más legible y comprensible para los desarrolladores.

## Mejora la escalabilidad del código

A medida que el código crece, mantener su integridad y extenderlo se vuelve más manejable gracias a las herramientas proporcionadas por TypeScript.

# MySQL

Es un sistema de gestión de bases de datos relacional de código abierto. Es uno de los sistemas de gestión de bases de datos más populares, conocido por su confiabilidad, facilidad de uso y rendimiento.

Utiliza una estructura de datos relacional para almacenar y organizar la información. Los datos se almacenan en tablas que están interrelacionadas mediante claves primarias y foráneas.

Utiliza SQL para realizar consultas y manipular datos. SQL es un estándar para el manejo de bases de datos relacionales.

Es conocido por su capacidad de escalar y manejar grandes volúmenes de datos, así como por su rapidez y eficiencia en el acceso a la información.

Es compatible con muchos sistemas operativos, lo que lo hace versátil y accesible para desarrolladores en diferentes entornos.

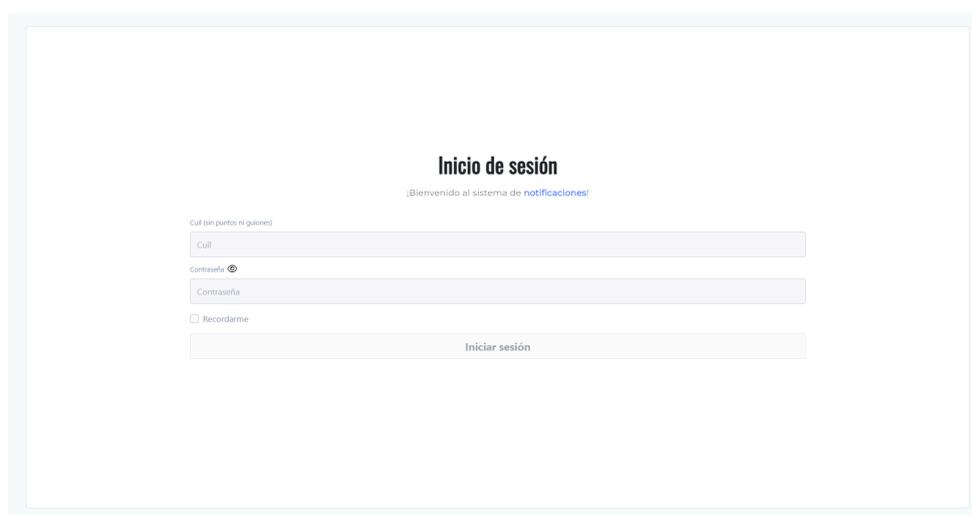
Es compatible con varios lenguajes de programación, lo que facilita la integración en aplicaciones desarrolladas en diferentes entornos.

Está disponible bajo la licencia de código abierto de GNU, lo que significa que es gratuito para su uso y también permite a los desarrolladores modificar el código fuente según sus necesidades.

# Implementación

## Inicio de sesión

Los agentes pueden acceder al sistema ingresando dos campos obligatorios. Los campos requeridos para ingresar son CUIL y contraseña. El CUIL se encuentra conformado por 11 caracteres numéricos y la contraseña (en la base de datos se encuentra encriptada<sup>3</sup>) es la seleccionada por el agente cuando creó su usuario. El agente puede almacenar en el LocalStorage<sup>4</sup> del navegador su CUIL para que en futuros logueos no deba tipearlo, si es que así lo desea. En caso de ingresar correctamente ambos campos podrá ingresar al sistema y se almacena en el LocalStorage el token generado. En caso contrario se le muestra un mensaje de credenciales incorrectas.



The screenshot shows a login form with the following elements:

- Title: **Inicio de sesión**
- Subtitle: (Bienvenido al sistema de [notificaciones!](#))
- Input field: CUIL (sin puntos ni guiones)
- Input field: Contraseña (with an eye icon for visibility toggle)
- Checkbox: Recordarme
- Button: Iniciar sesión

## Cambio de pestaña / Navegación

A los agentes se les muestra una lista de cinco ítems en la barra lateral izquierda. Cada ítem indica con un icono representativo y con un título el tema que se trata en cada pestaña. Además, aparece una descripción pequeña indicando que se puede realizar en cada una de ellas.

---

<sup>3</sup> Proceso de codificación de la información que convierte la representación original de la información en una forma alternativa conocida como texto cifrado.

<sup>4</sup> LocalStorage es una propiedad que tiene la función de almacenar datos de manera local en el navegador.

## Notificaciones

### ☰ Lista notificaciones

Liste sus notificaciones enviadas.

### ✉ Notificar externos

Envíe mails a externos.

### 👤 Externos

Administre externos.

### 👥 Grupos

Administre grupos.

### 📄 Plantillas

Administre plantillas.

## Listado de externos

Los agentes pueden visualizar una tabla con el listado de usuarios externos que se crearon en el sistema. En esta sección se administran los externos.

Al momento de realizar el llamado a la API se visualiza la tabla deshabilitada junto con un snippet y el título “Cargando listado” indicando que se están esperando los datos. Una vez obtenidos los datos de los externos la tabla muestra los resultados obtenidos, ya se encuentra habilitada y el título cambia a “Listado”.

En la parte superior aparecen dos botones. Un botón permite cargar nuevamente los datos de los usuarios externos y otro botón que permite crear un usuario externo.

Los datos que se visualizan son nombre y apellido, CUIL, dirección personal, dirección de trabajo, correo, teléfono y acciones. Dentro de las acciones los agentes pueden eliminar y modificar a los externos.

Se agregaron inputs que permiten a los agentes buscar usuarios externos en específico. Pueden buscar por nombre y apellido, CUIL, dirección personal, dirección de trabajo, correo y/o teléfono.

Por último, se muestra una leyenda en la parte inferior que indica la cantidad de usuarios externos que se están visualizando y la cantidad de usuarios externos que existen en el sistema.

### Externos

#### Listado

+ ↻

Nombre	Cuil	Dir. personal	Dir. trabajo	Correo	Telefono	Acciones
<input type="text" value="usuario"/>	<input type="text" value="Buscar por cuil"/>	<input type="text" value="Buscar por direccion pe"/>	<input type="text" value="Buscar por direccion de"/>	<input type="text" value="Buscar por correo"/>	<input type="text" value="Buscar por telefono"/>	
Prueba, Usuario	11111111111	Direccion Personal de prueba	Direccion larobal de prueba	usuarioprueba@gmail.com	2920111111	<span>✎</span> <span>✖</span>

Mostrando 1 al 1 de 1 usuarios externos << < 1 > >> 10 ▾

## Alta de usuario externo

Los agentes tienen la posibilidad de dar de alta usuarios externos. Para poder dar de alta un usuario externo es necesario completar diversos campos. Los campos obligatorios para crearlos son nombre, apellido, CUIL, DNI y correo. Los campos opcionales son teléfono, dirección personal y dirección laboral. Además, se tiene un checkbox que los agentes pueden tildar en el caso de que deseen enviarle un correo al externo creado para indicarles que fue dado de alta en el sistema. Al final se visualiza un botón que permite realizar la llamada a un endpoint de la API que permite crearlos y almacenarlos en la base de datos. Este botón se encuentra deshabilitado hasta que el agente no complete los campos obligatorios.

En el caso de que un agente ingrese un usuario externo cuyo CUIL y/o correo se encuentren en la base de datos se le indicará mediante un cartel (dialog) dicha situación.



Creación de usuario externo

Nombre Usuario ✓	Apellido Externo ✓
Cuil 11111111111 ✓	Dni 11111111 ✓
Correo usuarioprueba@gmail.com ✓	Número de teléfono (opcional) 2920111111 ✓
Dirección personal (opcional) Direccion personal de prueba ✓	Dirección laboral (opcional) Direccion laboral de prueba ✓

Notificar por correo al usuario externo dado de alta

[Crear usuario externo](#)

## Modificación de usuario externo

Los agentes tienen la posibilidad de modificar usuarios externos. Para poder modificar un usuario externo es necesario completar diversos campos que se autocompletan con la información del agente que lo dio de alta. Los campos obligatorios para modificarlos son nombre, apellido, CUIL, DNI y correo. Los campos opcionales son teléfono, dirección personal y dirección laboral. Al final se visualiza un botón que permite realizar la llamada a un endpoint de la API que permite modificarlos y almacenarlos en la base de datos. Este botón se encuentra deshabilitado hasta que el agente no complete los campos obligatorios. En el caso de que un agente ingrese un CUIL y/o correo que se encuentren en la base de datos se le indicará mediante un cartel (dialog) dicha situación.

Modificando usuario Prueba, Usuario

Nombre: Usuario ✓

Apellido: Prueba ✓

Cuil: 11111111111 ✓

Dni: 11111111 ✓

Correo: usuarioprueba@gmail.com ✓

Número de teléfono (opcional): 2920111111 ✓

Dirección personal (opcional): Dirección Personal de prueba ✓

Dirección laboral (opcional): Dirección larobal de prueba ✓

Modificar usuario externo

## Eliminación de usuario externo

Los agentes tienen la posibilidad de eliminar usuarios externos. Para poder eliminar un usuario externo el usuario debe seleccionar el externo que desea eliminar y presionar el botón que dice “eliminar”.

¿Eliminar a Prueba, Usuario?

Eliminar

## Alta de grupo

Los agentes tienen la posibilidad de dar de alta grupos. Para poder dar de alta un grupo es necesario completar dos campos. Los campos obligatorios para crearlos son nombre y descripción.

Al final se visualiza un botón que permite realizar la llamada a un endpoint de la API que permite crearlos y almacenarlos en la base de datos. Este botón se encuentra deshabilitado hasta que el agente no complete los campos obligatorios.

Creación de grupo

Nombre: Grupo 1 ✓

Descripción: Descripción del grupo 1 ✓

Una vez creado el grupo, luego deberá agregar los usuarios pertenecientes al mismo.

Crear Grupo

## Listado de grupos

Los agentes pueden visualizar una tabla con el listado de grupos que se crearon en el sistema. En esta sección se administran los grupos.

Al momento de realizar el llamado a la API se visualiza la tabla deshabilitada junto con un snippet y el título “Cargando listado” indicando que se están esperando los datos. Una vez obtenidos los datos de los grupos la tabla muestra los resultados obtenidos, ya se encuentra habilitada y el título cambia a “Listado”.

En la parte superior aparecen dos botones. Un botón permite cargar nuevamente los datos de los grupos y otro botón que permite crear un grupo.

Los datos que se visualizan son nombre, descripción y fecha de creación. Dentro de las acciones los agentes pueden eliminar y modificar los grupos.

Se agregaron tres inputs que permiten a los agentes buscar grupos en específico. Pueden buscar solo por nombre, descripción o fecha de creación.

Por último, se muestra una leyenda en la parte inferior que indica la cantidad de grupos que se están visualizando y la cantidad de grupos que existen en el sistema.

The screenshot shows a web interface titled "Grupos". Below the title is a "Listado" section with a table. The table has columns for "Nombre", "Description", "Fecha creación", and "Acciones". Above the table are three search input fields: "Buscar por nombre", "Buscar por descripción", and "Buscar por fecha de creación". To the right of the table are two buttons: a plus sign (+) and a refresh icon (circular arrow). Below the table, there is a pagination bar that says "Mostrando 0 al 0 de 0 usuarios externos" and a dropdown menu showing "10".

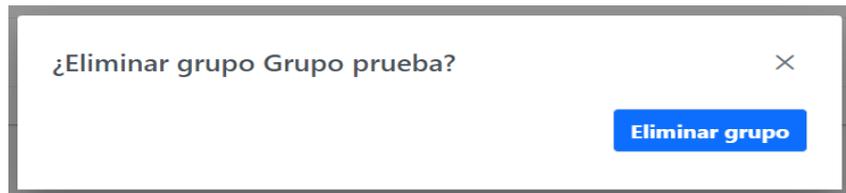
## Modificación de grupo

Los agentes tienen la posibilidad de modificar grupos. Para poder modificar un grupo es necesario completar dos campos que se autocompletan con la información del agente que lo dio de alta. Los campos obligatorios para modificarlos son nombre y descripción. Al final se visualiza un botón que permite realizar la llamada a un endpoint de la API que permite modificarlos y almacenarlos en la base de datos. Este botón se encuentra deshabilitado hasta que el agente no complete los campos obligatorios.

The screenshot shows a dialog box titled "Modificando grupo Grupo prueba". It has two input fields: "Nombre" with the value "Grupo prueba" and "Descripción" with the value "Grupo para realizar pruebas". Both fields have a green checkmark icon to their right. At the bottom right of the dialog is a blue button labeled "Modificar grupo".

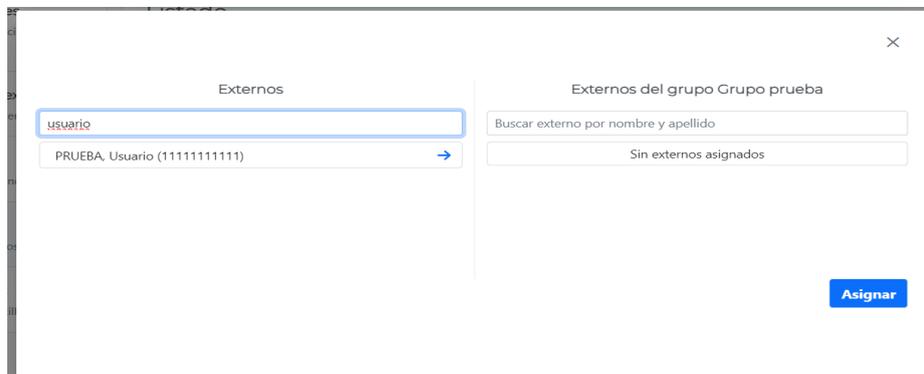
## Eliminación de grupo

Los agentes tienen la posibilidad de eliminar grupos. Para poder eliminar un grupo el agente debe seleccionar el que desea eliminar y presionar el botón que dice “Eliminar”.



## Asignación y designación de usuarios externos a grupos

Los agentes tienen la posibilidad de poder asignar usuarios externos a grupos. A los agentes se les muestra en la parte izquierda un listado de todos los usuarios externos que existen y aún no pertenecen al grupo seleccionado. Mientras, en la parte derecha se visualizan los usuarios externos pertenecientes al grupo seleccionado. El agente tiene la posibilidad de asignar o designar los usuarios externos que desee.



## Listado de plantillas

Los agentes pueden visualizar una tabla con el listado de plantillas que se crearon en el sistema. En esta sección se administran las plantillas.

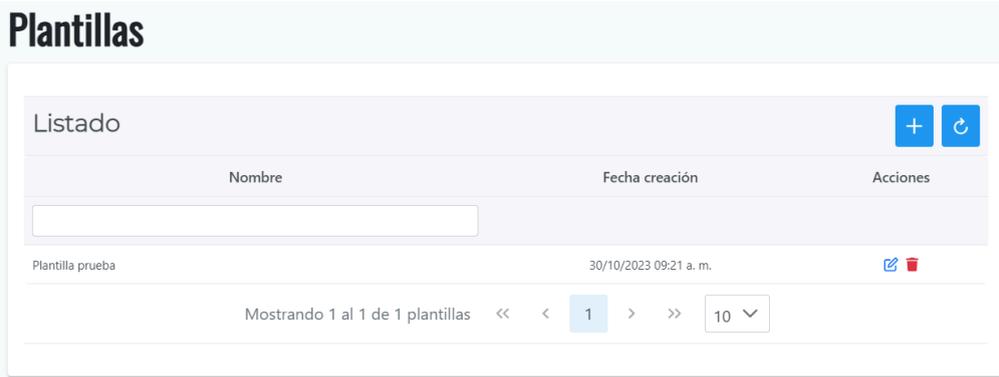
Al momento de realizar el llamado a la API se visualiza la tabla deshabilitada junto con un snippet y el título "Cargando listado" indicando que se están esperando los datos. Una vez obtenidos los datos de las plantillas la tabla muestra los resultados obtenidos, ya se encuentra habilitada y el título cambia a "Listado".

En la parte superior aparecen dos botones. Un botón permite cargar nuevamente los datos de las plantillas y otro botón que permite crear una plantilla.

Los datos que se visualizan son nombre y fecha de creación. Dentro de las acciones los agentes pueden eliminar y modificar las plantillas.

Se agregó un input que permite a los agentes buscar plantillas en específico. Pueden buscar solo por nombre.

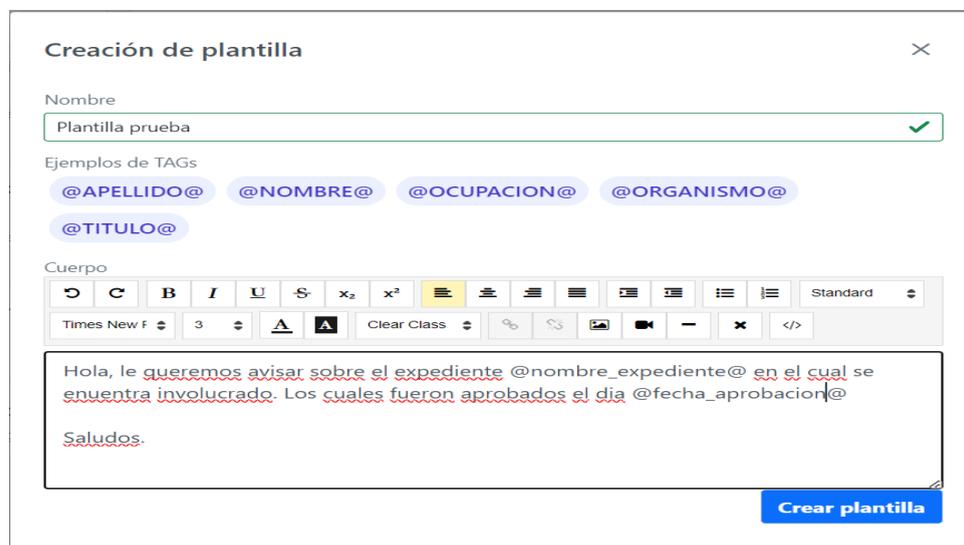
Por último, se muestra una leyenda en la parte inferior que indica la cantidad de plantillas que se están visualizando y la cantidad de plantillas que existen en el sistema.



## Alta de plantilla

Los agentes tienen la posibilidad de dar de alta plantillas. Para poder dar de alta una plantilla es necesario completar dos campos. Los campos obligatorios para crearlos son nombre y cuerpo. Varias plantillas pueden tener el mismo nombre y el cuerpo es el “body” de los correos. El body tiene la característica que pueden tener campos dinámicos denominados TAGs que se completan cuando se envía un correo. Se muestra un mensaje indicando ejemplos de TAGs, tienen la característica de que lo que se encuentra dentro de dos “@” va a ser sustituido en un futuro por los agentes cuando envíen el correo.

Al final se visualiza un botón que permite realizar la llamada a un endpoint de la API que permite crearlos y almacenarlos en la base de datos. Este botón se encuentra deshabilitado hasta que el agente no complete los campos obligatorios.



## Modificación de plantilla

Los agentes tienen la posibilidad de modificar plantillas. Para poder modificar una plantilla es necesario completar dos campos que se autocompletan con la información del agente que lo dio de alta. Los campos obligatorios para modificarlos son nombre y cuerpo. Al final se visualiza un botón que permite realizar la llamada a un endpoint de la API que permite modificarlos y almacenarlos en la base de datos. Este botón se encuentra deshabilitado hasta que el agente no complete los campos obligatorios.

Modificando plantilla Plantilla prueba

Nombre  
Plantilla prueba ✓

Ejemplos de TAGs  
@APELLIDO@ @NOMBRE@ @OCUPACION@ @ORGANISMO@ @TITULO@

Cuerpo

Hola, le queremos avisar sobre el expediente @nombre\_expediente@ en el cual se encuentra involucrado. Los cuales fueron aprobados el día @fecha\_aprobacion@.

Saludos.

Modificar plantilla

## Eliminación de plantilla

Los agentes tienen la posibilidad de eliminar plantillas. Para poder eliminar una plantilla el usuario debe seleccionar la que desea eliminar y presionar el botón que dice “Eliminar”.

¿Eliminar plantilla Plantilla prueba?

Eliminar

## Envío de notificación

El agente puede enviar notificaciones a usuarios externos. Para ello, debe completar diversos campos que son obligatorios.

En un principio debe completar el asunto que se le enviará por correo a los usuarios externos.

A continuación, debe seleccionar una plantilla de todas las plantillas que se le listen que pertenecen al grupo de trabajo. Además, junto a la plantilla tendrá la posibilidad de incluir su firma QR. Y, por último, completar los TAGs que posee la plantilla con la información correspondiente.

Debe seleccionar los usuarios externos a los que el agente desea enviar el correo. Como mínimo debe seleccionar un usuario y no hay límite máximo.

La última información que debe agregar son archivos adjuntos. Los archivos adjuntos son PDFs y/o imágenes (JPEG, JPG, PNG). Estos son necesarios porque sirven para apoyar a los expedientes o documentos de un expediente que el agente debe seleccionar para notificar al usuario externo. Los expedientes tienen filtros para buscar el deseado. Los agentes pueden buscar por número, fecha, año, clase y/o tema.

Por otro lado, los documentos de un expediente se visualizan en una tabla. Se presentan inputs que sirven como filtro para buscar el documento deseado. Se puede filtrar por tipo, número, año, descripción, clase, fecha, tema y/o área. Al final se muestra una leyenda indicando la cantidad de documentos que existen para ese expediente y la cantidad de documentos que se están visualizando.

A partir de este punto, si el agente asigna todos los campos obligatorios, se habilita el botón “Revisar notificación” que abre un modal que permite visualizar todos los datos ingresados de forma resumida. Esto permite verificar que no haya ningún error en la notificación y correo.

Si el agente está de acuerdo con la información que ingresó, debe hacer clic en el botón que dice “Enviar notificación”. El mismo notifica a los usuarios externos enviándoles un correo electrónico.

### Notificar externos

Asunto

Seleccionar plantilla ✔

Incluir firma en la plantilla



**Requerimiento** Plantilla Nueva Nueva con tags repetidos Plantilla prueba Plantilla prueba

Señor @APELLIDO@  
Esto es un requerimiento ....

**Editar plantilla**

---

#### Seleccionar grupo/s

Seleccione un grupo

Externos	Externos a notificar

**Seleccionar externo/s**

🔍 Buscar externo por nombre y apellido

Externos	Externos a notificar

**Seleccionar documento/s** ✔

0 archivos adjuntos

**Lista de expedientes**

	Número	Año	Fecha	Clase	Tema
<input type="checkbox"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="checkbox"/>		2022	23/08/22	TRIB.CTAS	*Adquisición de

Mostrando 1 al 1 de 1 expedientes << < 1 > >>

**Documentos del expediente**

\*Adquisición de \$:-

	Tipo	Número	Año	Descripción	Clase	Fecha	Tema	Area
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>	2022	<input type="text"/>				

Mostrando 1 al 1 de 1 documentos << < 1 > >>

[Cerrar](#)

Ocultar detalle
Enviar notificación a revisión
Revisar notificación

**Asunto**

Prueba de notificación

**Plantilla seleccionada**

Plantilla prueba (Firma incluida)

Hola, le queremos avisar sobre el expediente @nombre\_expediente@ en el cual se encuentra involucrado. Los cuales fueron aprobados el día @fecha\_aprobacion@.

Saludos.

**TAGs**

nombre\_expediente: expediente prueba

fecha\_aprobacion: 31/10/2023

**Destinatarios**

PRUEBA, Usuario

**Documentos**

213-183-2022 (APROBAR el Concurso de Pr...),

99-305-2022 (Aprobacion Concurso de pr...)

Enviar notificación

## Envío de notificación a revisión

El agente puede enviar notificaciones para revisión. Esto consiste en poder seleccionar a otro agente para que verifique que la notificación a enviar se encuentre con los datos correctos.

El agente puede enviar notificaciones a usuarios externos. Para ello, debe completar diversos campos que son obligatorios.

En un principio debe completar el asunto que se le enviará por correo a los usuarios externos. A continuación, deberá seleccionar una plantilla de todas las plantillas que se le listen que pertenecen al grupo de trabajo. Además, junto a la plantilla tendrá la posibilidad de incluir su firma QR. Y, por último, completar los TAGs que posee la plantilla con la información correspondiente.

Deberá seleccionar los usuarios externos a los que el agente desea enviar el correo. Como mínimo debe seleccionar un usuario y no hay límite máximo.

La última información que deberá agregar son archivos adjuntos. Los archivos adjuntos son PDFs y/o imágenes (JPEG, JPG, PNG). Estos son necesarios porque sirven para apoyar a los expedientes o documentos de un expediente que el agente debe seleccionar para notificar al usuario externo. Los expedientes tienen filtros para buscar el deseado. Los agentes pueden buscar por número, fecha, año, clase y/o tema.

Por otro lado, los documentos de un expediente se visualizan en una tabla. Se presentan inputs que sirven como filtro para buscar el documento deseado. Se puede filtrar por tipo, número, año, descripción, clase, fecha, tema y/o área. Al final se muestra una leyenda que indica la cantidad de documentos que existen para ese expediente y la cantidad de documentos que se están visualizando.

A partir de este punto, si el agente asigna todos los campos obligatorios, se habilita el botón "Enviar notificación a revisión". El cual abre un modal que permite visualizar todos los agentes que se pueden seleccionar para que la revisen. Solo pueden seleccionar un revisor. Y se tienen dos inputs que se pueden utilizar para filtrar por nombre y apellido o CUIL. Una vez seleccionado el revisor el agente debe hacer clic en el botón "Enviar a revisión".

### Notificar externos

Asunto

Seleccionar plantilla 

Incluir firma en la plantilla



**Requerimiento** Plantilla Nueva Nueva con tags repetidos Plantilla prueba Plantilla prueba

Señor @APELLIDO@

Esto es un requerimiento ...

[Editar plantilla](#)

### Seleccionar grupo/s

Seleccione un grupo ▼

Externos

>

>>

<

<<

Externos a notificar

### Seleccionar externo/s

Externos

>

>>

<

<<

Externos a notificar

### Seleccionar documento/s ✔

O archivos adjuntos

▼

Lista de expedientes

	Número	Año	Fecha	Clase	Tema	
<input type="checkbox"/>	<input type="text"/>		<input type="text"/>	<input type="text"/>	<input type="text"/>	
<input type="checkbox"/>		2022	23/08/22	TRIB.CTAS	*Adquisición de	

Mostrando 1 al 1 de 1 expedientes << < 1 > >>

### Documentos del expediente

\*Adquisición de

	Tipo	Número	Año	Descripción	Clase	Fecha	Tema	Area
<input type="checkbox"/>	...	<input type="text"/>						
			2022					

Mostrando 1 al 1 de 1 documentos << < 1 > >>

[Cerrar](#)

Ocultar detalle

Enviar notificación a revisión

Revisar notificación

Seleccione una persona para que sea revisor de la notificación

Nombre y apellido ↑↓	Cuil ↑↓
<input type="text" value="PRUEBA"/>	<input type="text"/>
<input checked="" type="checkbox"/> PRUEBA, Usuario	11111111111
<input type="checkbox"/> PRUEBAUNO, PruebaUNO	98989898989
<input type="checkbox"/> PRUEBADOS, PruebaDOS	56565656565
<input type="checkbox"/> PRUEBATRES, PruebaTRES	45454545454

PRUEBA, Usuario (X)

Enviar a revisión

Cerrar

## Listado notificaciones para revisión

A los agentes se les presenta una tabla con una lista de notificaciones que se le asignaron para que revisen. Los mismos pueden revisarlos haciendo clic en “Revisar”.

Notificaciones para su revisión

Enviada a revisión por	Asunto	Opciones
	notificaion prueba	<a href="#">Revisar</a>

## Revisar y aprobar notificación

El agente revisor podrá visualizar los datos que asignó el agente que lo mandó a revisar. Podrá verificar que los datos sean correctos. En caso contrario puede modificar algunos campos. Si los datos están correctos o el revisor terminó de modificarlos, debe primero almacenar los cambios haciendo clic en “Guardar como borrador”. Si desea enviar la notificación y los correos debe hacer clic en “Aprobar y enviar”.

### Revisión de notificación

Asunto 

notificaion prueba 

Seleccionar plantilla



Plantilla prueba

Hola, le queremos avisar sobre el expediente @nombre\_expediente@ en el cual se encuentra involucrado. Los cuales fueron aprobados el día @fecha\_aprobacion@.

Saludos.



Tribunal de cuentas de Río Negro

Editar plantilla

TAGs

nombre\_expediente

fecha\_aprobacion

Remitentes seleccionados

PRUEBA, Usuario
-----------------

Sin expedientes seleccionados

Documentos seleccionados

Guardar como borrador | Aprobar y enviar

## Listar notificaciones enviadas

Los agentes pueden visualizar las notificaciones y correos enviados en una tabla. En la misma se muestran estado, asunto, fecha, plantilla, destinatarios (usuarios externos), documentos, archivo original y opciones.

Los agentes pueden visualizar la lista de usuarios externos que se les envió el correo electrónico sobre esta notificación. Se muestra nombre y apellido, fecha de notificación, estado y comprobante (Archivo PDF que indica cuando se notificó el usuario externo). Se tienen 3 inputs que sirven como filtro para buscar un externo en particular. Los filtros que se utilizan son nombre y apellido, fecha de notificación y estado.

Los agentes, además, pueden visualizar en un listado los documentos por los cuales los usuarios externos fueron notificados. Al hacer clic sobre el icono con forma de ojo se les muestra el archivo PDF del documento.

## Listado de notificaciones

Notificaciones enviadas

Estado	Asunto	Fecha	Plantilla	Remitentes	Documentos	Archivo original	Opciones
En revisión	notificacion prueba	31/10/23, 10:09	Plantilla prueba	2	1		+

Nombre y Apellido	Fecha notificación	Estado	Comprobante
MARRONI, Juan Pablo	31/10/23, 10:09 a. m.	Programado	
PRUEBA, Usuario	31/10/23, 10:09 a. m.	Programado	

Documento	Visualizar
213/2022/183	

## Postman

### Vista general

El nombre del proyecto creado en Postman es “Notifications (CHASQUI)”. Se crearon carpetas que se consideraron necesarias para agrupar el conjunto de endpoints que tengan relación entre ellos. Por ejemplo, la carpeta con el nombre “Rutas de Usuarios” contiene endpoints para crear, eliminar, modificar y obtener usuarios externos. Se dejaron endpoints sin carpetas ya que tenían una funcionalidad diferente al resto de endpoints.

Los métodos GET, POST, PUT y DELETE fueron los utilizados para los diferentes endpoints de la aplicación.

### Variables de entorno

Actualmente, existen en Postman 22 variables de entorno, que fueron utilizadas para configurar y controlar el comportamiento de la aplicación. Las variables de entorno permitieron ajustar la configuración de la aplicación sin necesidad de modificar el código fuente. Aportaron seguridad porque fueron utilizadas para almacenar información sensible, como contraseñas y claves de API de forma más segura, ya que no se almacenan directamente en el código fuente. Esto evitó exponer información accidentalmente en los repositorios de código. Por último, permitieron adaptar el comportamiento de la aplicación según nuestras preferencias o necesidades. Por ejemplo, se pudo establecer en ellas la URL donde se encuentra el backend o la ruta donde se encuentran almacenados los documentos.

### Métodos de usuarios externos

Véase anexo.

## Jira

Sprint 0 - Product Backlog

NAD-1 Inicio de sesión del agente	TAREAS POR HACER	
NAD-2 Listado de usuarios externos	TAREAS POR HACER	
NAD-3 Alta de usuario externo	TAREAS POR HACER	
NAD-4 Modificación de usuario externo	TAREAS POR HACER	
NAD-5 Eliminación de usuario externo	TAREAS POR HACER	
NAD-6 Listado de grupos	TAREAS POR HACER	
NAD-7 Alta de grupo	TAREAS POR HACER	
NAD-8 Modificación de grupo	TAREAS POR HACER	
NAD-9 Eliminación de grupo	TAREAS POR HACER	
NAD-10 Asignación y designación de usuarios externos a un grupo	TAREAS POR HACER	
NAD-11 Listado de plantillas	TAREAS POR HACER	
NAD-12 Alta de plantilla	TAREAS POR HACER	
NAD-13 Modificación de plantilla	TAREAS POR HACER	
NAD-14 Eliminación de plantilla	TAREAS POR HACER	
NAD-15 Envío de notificación	TAREAS POR HACER	
NAD-16 Envío de notificación a revisión	TAREAS POR HACER	
NAD-17 Listado de notificaciones	TAREAS POR HACER	
NAD-18 Listado de notificación a revisión	TAREAS POR HACER	
NAD-19 Revisar y aprobar notificación	TAREAS POR HACER	

# Sprint 1

Tablero Sprint 1		0	0	Iniciar sprint
NAD-1 Inicio de sesión del agente	TAREAS POR HACER			
NAD-2 Listado de usuarios externos	TAREAS POR HACER			
NAD-3 Alta de usuario externo	TAREAS POR HACER			
NAD-4 Modificación de usuario externo	TAREAS POR HACER			
NAD-5 Eliminación de usuario externo	TAREAS POR HACER			

# Sprint 1 - Kanban

## Tablero Sprint 1

POR HACER 5

Inicio de sesión del agente

NAD-1

Listado de usuarios externos

NAD-2

Alta de usuario externo

NAD-3

Modificación de usuario externo

NAD-4

Eliminación de usuario externo

NAD-5

EN CURSO

TESTING ✓

PRODUCTION

## Conclusiones

La implementación del sistema "Chasqui" en el Tribunal de Cuentas de Río Negro ha transformado significativamente los procesos de comunicación y gestión de notificaciones entre los agentes y los externos. La situación inicial, que caracterizada por la dispersión de datos y la falta de un canal unificado, se resolvió mediante la centralización de la información en una aplicación web, lo que permitió una gestión más eficiente y organizada.

El sistema no solo mejoró la velocidad de las notificaciones, sino que también redujo los errores y tiempos de respuesta, incrementando la eficiencia de los procesos internos. La digitalización de los documentos facilitó el acceso a la información y permitió a los agentes y externos interactuar de forma más ágil, tanto de manera presencial como remota.

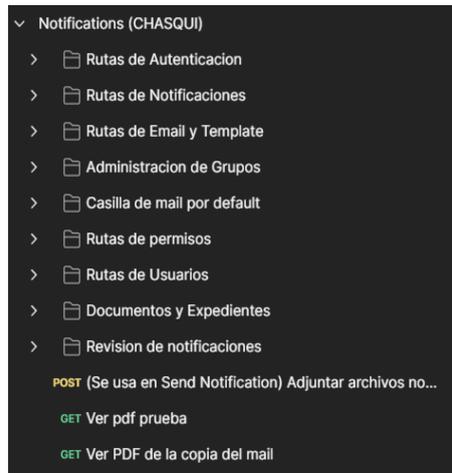
La validación del sistema se llevó a cabo mediante el monitoreo de las notificaciones enviadas, el acceso a los archivos adjuntos y la comprobación de recepción por parte de los externos. El respaldo periódico de la información garantizó la conservación de los datos y la reducción de riesgos en caso de pérdidas accidentales.

Las mejoras introducidas, como la automatización de tareas repetitivas, la facilidad de búsqueda y la posibilidad de acceder a un historial de notificaciones, beneficiaron tanto a los agentes como a los externos al reducir la carga administrativa y optimizar el flujo de trabajo. En conjunto, el sistema "Chasqui" representó una mejora significativa en la gestión institucional, haciendo que los procesos sean más transparentes, rápidos y efectivos, lo que contribuyó a la mejora global del servicio ofrecido por el Tribunal.

# REFERENCIAS BIBLIOGRÁFICAS

- [1] *Metodología Scrum* | Trigas Gallego, Manuel.  
<https://openaccess.uoc.edu/handle/10609/17885>
- [2] *Formularios de Google: Generador de formularios en línea* | Google.  
[https://www.google.com/intl/es-419\\_ar/forms/about/](https://www.google.com/intl/es-419_ar/forms/about/)
- [3] *Bienvenido a Jira software* | Atlassian.  
<https://www.atlassian.com/es/software/jira/guides/getting-started/introduction>
- [4] *Documentation for Visual Studio Code* | Microsoft.  
<https://code.visualstudio.com/>
- [5] *Introduction to the Angular docs* | Google.  
<https://angular.io/>
- [6] *HTML: Lenguaje de etiquetas de hipertexto* | Mozilla.  
<https://developer.mozilla.org/es/docs/Web/HTML>
- [7] *¿Qué es CSS?* | Mozilla.  
[https://developer.mozilla.org/es/docs/Learn/CSS/First\\_steps/What\\_is\\_CSS](https://developer.mozilla.org/es/docs/Learn/CSS/First_steps/What_is_CSS)
- [8] *TypeScript documentation* | Microsoft.  
<https://www.typescriptlang.org/>
- [9] *The PHP framework for web artisans* | LaravelLLC.  
<https://laravel.com/>
- [10] *MySQL documentation* | Oracle.  
<https://www.mysql.com/>
- [11] *TablePLUS documentation* | TablePLUS Inc.  
<https://tableplus.com/>
- [12] *What is Postman?* | Postman Inc.  
<https://www.postman.com/>

# Anexo



Variable	Type	Initial value	Current value
<input checked="" type="checkbox"/> token	default	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiS...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUliwia2kiIiA6ICIzch...
<input checked="" type="checkbox"/> backend_url	default	http://	http:// /api
<input checked="" type="checkbox"/> keycloak_url	default	http://	http://1
<input checked="" type="checkbox"/> secret_id	default	4836af83-4ecc-48d2-91cb-aafbe8...	4836af83-4ecc-48d2-91cb-aafbe80a75d3
<input checked="" type="checkbox"/> client_id	default		
<input checked="" type="checkbox"/> intranet_backend_url	default	http://	http:// /api
<input checked="" type="checkbox"/> workflow_webservice_url	default	http://	http:// /web-service
<input checked="" type="checkbox"/> workflow_backend_url	default	http://	http:// /api
<input checked="" type="checkbox"/> notifications_backend_url	default	http://	http:// /api
<input checked="" type="checkbox"/> workflow_token	default	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiS...	eyJhbGciOiJSUzI1NiIsInR5cCIgOiAiSldUliwia2kiIiA6ICIzch...
<input checked="" type="checkbox"/> people_backend_url	default		
<input checked="" type="checkbox"/> ebla_backend	default	http://	http:// /api
<input checked="" type="checkbox"/> facade_ebla_url	default	http://	http:// /ebla
<input checked="" type="checkbox"/> ebla_backend_url	default	http://	http:// /api
<input checked="" type="checkbox"/> security_url	default	http://	http:// /api
<input checked="" type="checkbox"/> external_notification_backend	default	http://	http:// /api
<input checked="" type="checkbox"/> files_service	default	http://	http:// 2/api
<input checked="" type="checkbox"/> api_facade_token	default	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1...	eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiIsInR5cCIgOiAiSldUliwia2kiIiA6ICIzch...
<input checked="" type="checkbox"/> external_notifications_url	default	http://	http:// /api
<input checked="" type="checkbox"/> people_id	default	134	134
<input checked="" type="checkbox"/> facade_url	default	http://	http:// /api
<input checked="" type="checkbox"/> accessToken	default		

Aclaración: Las palabras que se encuentran entre “{{” y “}}” hacen referencia a las variables de entorno.

## Obtener externos

Método

**GET**

URL

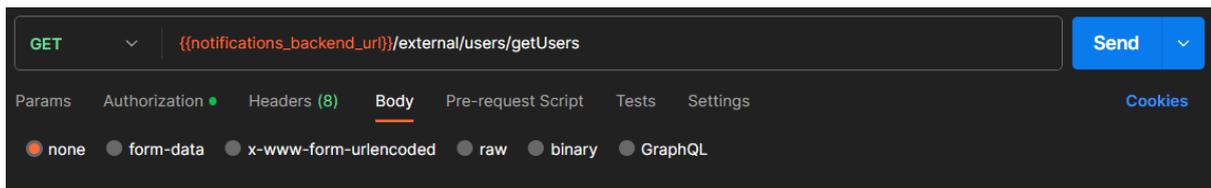
**{{notifications\_backend\_url}}/external/users/getUsers**

## Encabezados

- Cache-Control: no-cache
- Content-Type: application/x-www-form-urlencoded
- Accept: \*/\*
- Bearer token: **{{token}}**
- AuthorizationWorkflow: **{{workflow\_token}}**

## Parámetros

Sin parámetros.



## Mensaje de respuesta con parámetros que pasen la validación

```
[
  {
    "id": 512,
    "CUIL": "11111111111",
    "DNI": 11111111,
    "name": "Usuario",
    "lastname": "PRUEBA",
    "lastname_name": "PRUEBA, Usuario",
    "personal_address": "Direccion Personal de prueba",
    "work_address": "Direccion larobal de prueba",
    "email": "usuarioprueba@gmail.com",
    "phone": "2920111111",
    "blocked": 0,
    "created_at": "2023-09-14T16:32:19.000000Z",
    "updated_at": "2023-10-30T11:42:47.000000Z",
    "deleted_at": null
  },
]
```

## Datos obtenidos

Se obtiene un arreglo de usuarios externos creados en el sistema. Los datos obtenidos de cada externo son id, CUIL, DNI, name, lastname, lastname\_name, personal\_adress, work\_adress, email, phone, blocked, created\_at, updated\_at y deleted\_at.

## Código de estado

**200 OK**

## Alta de usuario externo

Método

**POST**

URL

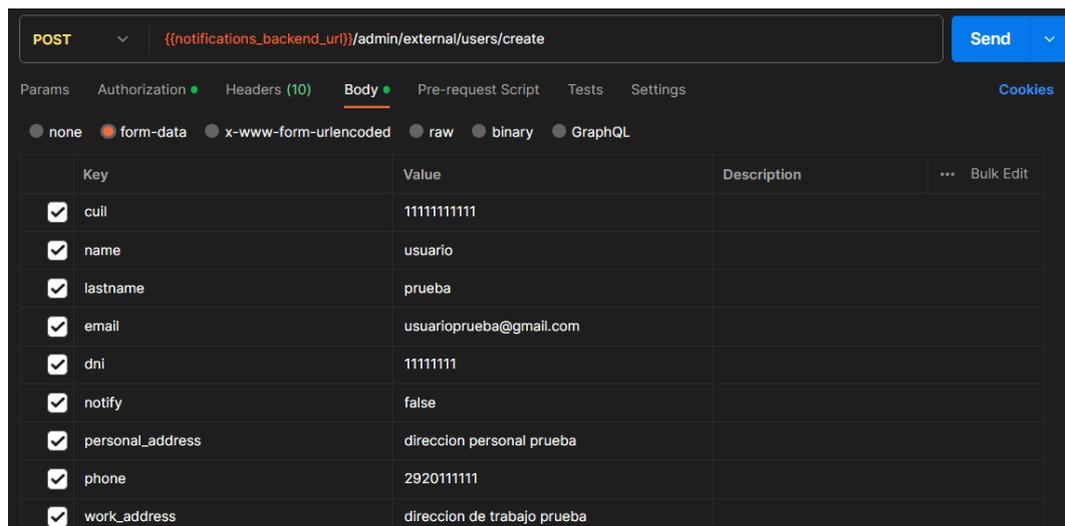
**{{notifications\_backend\_url}}/admin/external/users/create**

Encabezados

- Cache-Control: no-cache
- Content-Type: application/x-www-form-urlencoded
- Accept: \*/\*
- Bearer token: **{{token}}**
- AuthorizationWorkflow: **{{workflow\_token}}**

Parámetros

CUIL, name, lastname, email, DNI, notify, personal\_address, phone y work\_address.



The screenshot shows a REST client interface with a POST request to the URL `{{notifications_backend_url}}/admin/external/users/create`. The 'Body' tab is selected, and the data is formatted as form-data. The following table represents the data shown in the interface:

Key	Value	Description
<input checked="" type="checkbox"/> cuil	1111111111	
<input checked="" type="checkbox"/> name	usuario	
<input checked="" type="checkbox"/> lastname	prueba	
<input checked="" type="checkbox"/> email	usuarioprueba@gmail.com	
<input checked="" type="checkbox"/> dni	11111111	
<input checked="" type="checkbox"/> notify	false	
<input checked="" type="checkbox"/> personal_address	direccion personal prueba	
<input checked="" type="checkbox"/> phone	292011111	
<input checked="" type="checkbox"/> work_address	direccion de trabajo prueba	

Mensaje de respuesta con parámetros que pasen la validación

```
{
  "message": "El usuario ha sido creado.",
  "user": {
    "created_at": "2023-10-31T11:11:57.000000Z",
    "CUIL": "1111111111",
    "DNI": "11111111",
    "email": "usuarioprueba@gmail.com",
    "id": 500,
    "lastname": "prueba",
    "lastname_name": "prueba, usuario",
    "name": "usuario",
```

```
"personal_address": "direccion personal prueba",
"phone": "2920111111",
"work_address": "direccion de trabajo prueba",
"updated_at": "2023-10-31T11:11:57.000000Z"
},
}
```

### Datos obtenidos

message: Mensaje que indica lo sucedido.

user: Usuario creado en la base de datos. El mismo devuelve created\_at, CUIL, DNI, email, id, lastname\_name, name, lastname, personal\_address, phone, work\_address y updated\_at.

Código de estado

**200 OK**

### Modificación de usuario externo

Método

**PUT**

URL

**{{notifications\_backend\_url}}/external/users/update**

Encabezados

- Cache-Control: no-cache
- Content-Type: application/x-www-form-urlencoded
- Accept: \*/\*
- Bearer token: **{{token}}**
- AuthorizationWorkflow: **{{workflow\_token}}**

Parámetros

CUIL, DNI, name, email, id, lastname, lastname\_name, name, personal\_address, phone y work\_address.

POST `{{notifications_backend_url}}/external/users/update` Send

Params Authorization Headers (10) **Body** Pre-request Script Tests Settings Cookies

none
  form-data
  x-www-form-urlencoded
  raw
  binary
  GraphQL

Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> cull	1111111111			
<input checked="" type="checkbox"/> dni	11111111			
<input checked="" type="checkbox"/> name	usuario			
<input checked="" type="checkbox"/> email	usuarioprueba@gmail.com			
<input checked="" type="checkbox"/> id	500			
<input checked="" type="checkbox"/> lastname	prueba			
<input checked="" type="checkbox"/> lastname_name	usuario, prueba			
<input checked="" type="checkbox"/> name	usuario			
<input checked="" type="checkbox"/> personal_address	Direccion personal prueba			
<input checked="" type="checkbox"/> phone	2920111111			
<input checked="" type="checkbox"/> work_address	Direccion de trabajo prueba			

Mensaje de respuesta con parámetros que pasen la validación

```
{
  "message": "Usuario modificado correctamente.",
}
```

Datos obtenidos

message: Mensaje que indica lo sucedido.

Código de estado

**200 OK**

Eliminación de usuario externo

Método

**DELETE**

URL

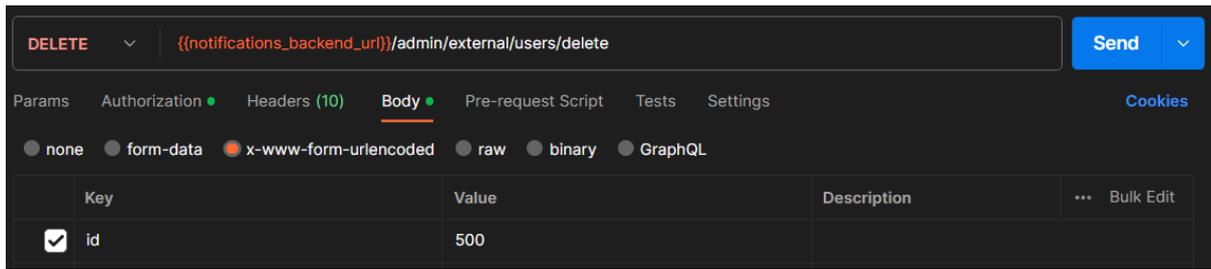
**`{{notifications_backend_url}}/admin/external/users/delete`**

Encabezados

- Cache-Control: no-cache
- Content-Type: application/x-www-form-urlencoded
- Accept: \*/\*
- Bearer token: `{{token}}`
- AuthorizationWorkflow: `{{workflow_token}}`

Parámetros

Id.



Mensaje de respuesta con parámetros que pasen la validación

```
{  
  "message": "Se elimino al usuario correctamente.",  
}
```

Datos obtenidos

message: Mensaje que indica lo sucedido.

Código de estado

**200 OK**

Mensaje de respuesta sin estar logueado en todos los endpoints

```
{  
  "error": {  
    "status": 401,  
    "title": "Error",  
    "description": "[Keycloak Guard] Expired token"  
  }  
}
```

Datos obtenidos

error: Objeto JSON que indica el error. Tiene tres propiedades: status, title y description.

status: Indica el código de estado de la solicitud.

title: Título del mensaje.

description: Descripción del error. En este caso el tiempo de vida del token expiró.

Código de estado

401 - Unauthorized

# Referencias

[1] Angular documentación.

<https://angular.io/docs>

[2] Laravel documentación.

<https://laravel.com/>

[3] Typescript documentación.

<https://www.typescriptlang.org/>

[4] Postman documentación.

<https://www.postman.com/>

[5] Jira documentación.

<https://www.atlassian.com/es/software/jira>

[6] Visual Studio Code documentación.

<https://code.visualstudio.com/>

[7] MySQL documentación.

<https://www.mysql.com/>