

Automatización de Requerimientos Fiscales

Propuesta de IT

Responsable:

Isaac Herrera Mareño

Nuevatel, 2023

ÍNDICE

1. Introducción.....	3
1.1. Antecedentes del Proyecto.....	3
2. Problema.....	4
2.1. Planteamiento del Problema.....	4
2.2. Formulación del Problema.....	4
3. Objetivos.....	5
3.1. General.....	5
3.2. Específico.....	5
4. Justificación.....	6
5. Alcances.....	6
5.1. Alcance Técnico.....	6
6. Marco Teórico.....	7
6.1. Conceptos Básicos.....	7
6.2. Modelos y Teorías Relacionadas.....	7
7. Ingeniería de Requerimientos.....	9
7.1. Requerimientos Funcionales.....	9
7.2. Requerimientos no Funcionales.....	10
8. Metodologías y herramientas de desarrollo.....	11
8.1. Metodología Scrum.....	11
8.2. Arquitectura Cliente Servidor.....	11
8.3. Herramientas utilizadas.....	12
9. Modelado de Software.....	13
9.1. Pruebas Unitarias.....	13
10. Resultados esperados.....	15
10.1. Evaluación de Desempeño.....	15
10.2. Casos de Éxito.....	15
11. Bibliografía.....	16
12. Anexos.....	18

1. Introducción

1.1. Antecedentes del Proyecto

El proyecto surge como una necesidad de automatizar el proceso de manejo de requerimientos fiscales que actualmente consume aproximadamente el 45% del tiempo del equipo de ITC. El proceso actual involucra múltiples etapas y plataformas, desde la recepción del requerimiento por parte de los abogados hasta la extracción de información necesaria.

2. Problema

2.1.Planteamiento del Problema

El manejo de requerimientos fiscales en el departamento de IT es un proceso altamente ineficiente. Los requerimientos llegan a través de la plataforma ServiceDesk, y la información necesaria se obtiene mediante consultas manuales en bases de datos. Esto no sólo es tedioso, sino que también es propenso a errores manuales y consume una gran cantidad de tiempo que podría utilizarse en tareas más críticas.

2.2.Formulación del Problema

¿Cómo se puede automatizar el proceso de manejo de requerimientos fiscales en el departamento de ITC de Nuevatel para mejorar la eficiencia y permitir que el equipo se enfoque en tareas más críticas?

3. Objetivos

3.1. General

Automatizar el proceso de manejo de requerimientos fiscales en el departamento de ITC para mejorar la eficiencia operativa y liberar recursos para tareas más críticas.

3.2. Específico

Desarrollar una aplicación que permita a los abogados de la empresa, ingresar requerimientos fiscales que se procesan automáticamente, para generar la información requerida. Reducir el tiempo dedicado por el equipo de ITC, pasando del 45% actual al 2%.

4. Justificación

La automatización del proceso permitirá un mejor uso de los recursos dentro del departamento de IT, permitiendo que el equipo se enfoque en tareas que aporten un mayor valor a la empresa. Además, al reducir el tiempo de respuesta para los requerimientos fiscales, se mejorará la eficiencia en toda la organización.

5. Alcances

5.1. Alcance Técnico

El proyecto implica el desarrollo de una aplicación web utilizando Node.js, Express.js, y MySQL. Se integrará con las bases de datos existentes para automatizar las consultas necesarias para cumplir con los requerimientos fiscales.

6. Marco Teórico

6.1. Conceptos Básicos

Automatización de Procesos: La automatización se ha convertido en un elemento clave en la modernización de diversas industrias. Va más allá de simplemente reemplazar tareas manuales con máquinas; implica el uso de tecnologías avanzadas como el aprendizaje automático y la inteligencia artificial para realizar tareas complejas de manera eficiente. La implementación de la automatización en el manejo de requerimientos fiscales es un paso significativo hacia la modernización del departamento de IT [1].

Gestión de Requerimientos: La gestión eficaz de requerimientos es crucial para cualquier proyecto de software. Esta práctica se centra no solo en recopilar y documentar los requisitos iniciales, sino también en adaptarse a los cambios y actualizaciones durante todo el ciclo de vida del proyecto. Este concepto es especialmente relevante en el contexto del proyecto, donde los requerimientos pueden cambiar en función de las regulaciones fiscales o las necesidades del departamento legal [2].

Node.js: Node.js es más que un simple entorno de ejecución; representa un cambio de paradigma en el desarrollo de aplicaciones web y de red. Su modelo de entrada/salida no bloqueante lo hace particularmente eficaz para aplicaciones que requieren alta concurrencia, como es el caso de este proyecto. Su ecosistema, el registro de paquetes npm, ofrece numerosas bibliotecas y marcos que aceleran el proceso de desarrollo [3].

6.2. Modelos y Teorías Relacionadas

Modelo de Desarrollo Ágil (Scrum): Scrum no es solo un marco para el desarrollo ágil; es una mentalidad que promueve la colaboración y adaptabilidad. En el entorno de constante cambio de los requerimientos fiscales y legales, un enfoque ágil permite al equipo adaptarse rápidamente a nuevas demandas y regulaciones [4].

Arquitectura Cliente-Servidor: En el modelo cliente-servidor, el servidor actúa como un centro de recursos y servicios que son consumidos por los clientes. Esta arquitectura

se destaca por su flexibilidad y escalabilidad, lo que la convierte en una opción ideal para sistemas que deben manejar un gran volumen de solicitudes, como es el caso del sistema propuesto para manejar requerimientos fiscales [5].

Teoría de la Carga Cognitiva: La Teoría de la Carga Cognitiva argumenta que la eficiencia en el aprendizaje y la realización de tareas se ven afectadas por la cantidad de recursos cognitivos requeridos. Al automatizar procesos, se busca liberar recursos cognitivos, permitiendo a los empleados enfocarse en tareas más significativas que no pueden ser automatizadas [6].

7. Ingeniería de Requerimientos

7.1.Requerimientos Funcionales

Consulta de Datos de Usuarios:

Sub-requerimiento 1: El sistema debe permitir la consulta de información básica de los usuarios, a partir de un identificador único como el nombre AD o el número de identificación.

Sub-requerimiento 2: Debe ser posible realizar búsquedas avanzadas utilizando múltiples criterios, como rango de fechas, tipos de datos a buscar, entre otros.

Sub-requerimiento 3: Los resultados de la consulta deben poder exportarse en varios formatos, incluidos texto y XLS.

Manejo de Auditorías Internas:

Sub-requerimiento 1: El sistema debe permitir el registro detallado de todas las operaciones realizadas, incluidas las consultas y generación de informes.

Sub-requerimiento 2: Debe ser posible consultar el historial de auditoría filtrado por usuario o tipo de operación.

Sub-requerimiento 3: El sistema debe notificar a los administradores en caso de actividades sospechosas o no autorizadas.

Interfaz de Usuario Amigable:

Sub-requerimiento 1: La interfaz debe incluir un panel de control desde donde los administradores puedan acceder a todas las funcionalidades del sistema.

Sub-requerimiento 2: Los formularios y pantallas de consulta deben diseñarse siguiendo las mejores prácticas de UX para garantizar una experiencia de usuario fluida.

Gestión de Accesos y Permisos:

Sub-requerimiento 1: Debe permitir la asignación de roles y permisos a diferentes usuarios para restringir el acceso a ciertas funcionalidades.

Sub-requerimiento 2: El sistema debe permitir la recuperación segura de contraseñas a través de un mecanismo de verificación de identidad.

7.2.Requerimientos no Funcionales

Seguridad: Todas las transacciones y consultas dentro del sistema deben ser seguras y cumplir con las políticas de seguridad de la empresa y las regulaciones legales [11].

Rendimiento y Escalabilidad: El sistema debe ser capaz de manejar un alto volumen de consultas y reportes diariamente, con la posibilidad de escalar según las necesidades de la empresa [12].

Disponibilidad: Dado que el sistema es crucial para las operaciones legales y fiscales, debe estar disponible las 24 horas del día, los 7 días de la semana [13].

Documentación y Mantenibilidad: El código fuente debe estar bien documentado para facilitar las actualizaciones y el mantenimiento a largo plazo [14].

8. Metodologías y herramientas de desarrollo

8.1. Metodología Scrum

La metodología Scrum ha sido la elegida para el desarrollo de este proyecto debido a su enfoque ágil y su adaptabilidad a cambios en los requisitos. Esta metodología divide el desarrollo en iteraciones llamadas "sprints", que suelen durar dos semanas. Cada sprint tiene un conjunto de tareas que se planifican durante la reunión de planificación del sprint y se revisan en la reunión de revisión del sprint.

Roles en Scrum: Los tres roles principales en un equipo Scrum son el Dueño del Producto, el Scrum Master y el Equipo de Desarrollo. El Dueño del Producto es responsable de la visión del proyecto y la priorización de las tareas. El Scrum Master facilita el proceso Scrum, y el Equipo de Desarrollo es responsable de completar las tareas.

Artefactos y Ceremonias: Los principales artefactos en Scrum son el Backlog del Producto y el Backlog del Sprint. Las ceremonias incluyen la planificación del sprint, la reunión diaria de Scrum, la revisión del sprint y la retrospectiva del sprint [19].

8.2. Arquitectura Cliente Servidor

El proyecto sigue una arquitectura Cliente-Servidor, donde el servidor es responsable de gestionar la base de datos y la lógica empresarial, y el cliente se encarga de la interfaz de usuario y la presentación de datos.

Capa de Servidor: Implementada usando Node.js, es responsable de la gestión de las consultas a la base de datos y la ejecución de la lógica de negocio.

Capa de Cliente: Construida con Handlebars para las plantillas y JavaScript puro para la lógica del lado del cliente, es responsable de presentar datos al usuario y manejar la interacción del usuario [20].

8.3.Herramientas utilizadas

Node.js: Utilizado para la implementación del servidor debido a su rendimiento y escalabilidad.

MySQL: Base de datos relacional utilizada para almacenar toda la información del sistema.

Handlebars: Motor de plantillas utilizado para generar las vistas HTML del cliente.

Express.js: Marco de aplicación web para Node.js, utilizado para construir la API y manejar las rutas del servidor.

Socket.io: Utilizado para la comunicación en tiempo real entre el cliente y el servidor [21].

9. Modelado de Software

9.1.Pruebas Unitarias

Las pruebas unitarias son esenciales para validar cada componente del software de manera individual, asegurando que cada parte funcione como se espera antes de integrarse en el sistema completo. Para este proyecto, se utilizará el framework de pruebas Jest para Node.js.

Consulta de Datos de Usuarios:

Prueba 1: Verificar que la consulta de información de un usuario retorne los datos correctos.

Prueba 2: Confirmar que la consulta de un usuario inexistente retorna un error específico.

Prueba 3: Asegurar que las consultas puedan manejarse simultáneamente sin afectar el rendimiento del sistema.

Manejo de Auditorías Internas:

Prueba 1: Comprobar que todas las operaciones en el sistema sean registradas en el historial de auditoría.

Prueba 2: Confirmar que las actividades sospechosas generan una alerta para los administradores.

Prueba 3: Validar que el historial de auditoría no pueda ser alterado o eliminado por usuarios no autorizados.

Interfaz de Usuario Amigable:

Prueba 1: Asegurar que todas las páginas carguen en un tiempo aceptable.

Prueba 2: Verificar que los tooltips y ayudas en línea se muestren correctamente.

Prueba 3: Confirmar que los formularios validan las entradas del usuario antes de enviar los datos.

Gestión de Accesos y Permisos:

Prueba 1: Validar que solo los usuarios autorizados puedan acceder a funciones restringidas.

Prueba 2: Asegurar que la recuperación de contraseña funcione de manera segura y efectiva.

Cada una de estas pruebas se desarrollará utilizando técnicas de mocking para simular las interacciones con bases de datos y otros componentes del sistema. Las pruebas se automatizan y formarán parte del pipeline de CI/CD para asegurar que cualquier cambio en el código no rompa la funcionalidad existente [22].

10. Resultados esperados

10.1. Evaluación de Desempeño

Tiempo a Ahorrar: Se espera que el departamento de ITC reduzca el tiempo dedicado a los Requerimientos Fiscales del 45% al 2%, permitiendo la redistribución de recursos a tareas más críticas.

Calidad de los Datos: Se anticipa una mejora significativa en la precisión de la generación de informes, minimizando los errores.

10.2. Casos de Éxito

Reducción de Carga de Trabajo: Se espera que la carga de trabajo para el equipo de ITC se reduzca significativamente, liberando recursos para otras tareas de mayor importancia.

Respuesta Rápida: Se prevé que la velocidad para atender los Requerimientos Fiscales mejorará dramáticamente, reduciendo el tiempo de respuesta de horas o días a minutos, dependiendo de la complejidad del requerimiento.

Auto-servicio para el Departamento Legal: La automatización de este proceso debería permitir que el departamento legal realice muchas de estas tareas sin depender del departamento de ITC, aumentando así su eficiencia.

11. Bibliografia

- Smith, J. (2018). The Basics of Process Automation. *Journal of Automation*.
- Davis, A. (2013). *Software Requirements: Objects, Functions, and States*. Prentice Hall.
- Tilkov, S., & Vinoski, S. (2010). Node.js: Using JavaScript to Build High-Performance Network Programs. *IEEE Internet Computing*, 14(6), 80-83.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide*. ScrumGuides.org.
- Tanenbaum, A. S., & Van Steen, M. (2007). *Distributed Systems: Principles and Paradigms*. Prentice-Hall.
- Sweller, J. (1988). Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2), 257-285.
- Robertson, S., & Robertson, J. (2012). *Mastering the Requirements Process: Getting Requirements Right*. 3rd ed., Addison-Wesley Professional. ISBN: 978-0321815743.
- Wieggers, K. E., & Beatty, J. (2013). *Software Requirements*. 3rd ed., Microsoft Press, USA. ISBN: 978-0735679665.
- Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional. ISBN: 978-0137043293.
- Ambler, S. W. (2002). *Introduction to Disciplined Agile Delivery: A Small Agile Team's Journey from Scrum to Continuous Delivery*. Disciplined Agile Consortium.
- OWASP. (2020). OWASP Top 10. Retrieved from <https://owasp.org/www-project-top-ten/>
- Bass, L., Clements, P., & Kazman, R. (2003). *Software Architecture in Practice*. 2nd ed., SEI Series in Software Engineering, Addison Wesley. ISBN: 978-0321154958.
- Jalote, P. (2002). *Software Project Management in Practice*. Addison-Wesley Professional. ISBN: 978-0201737219.

McConnell, S. (2004). Code Complete: A Practical Handbook of Software Construction. 2nd ed., Microsoft Press, USA. ISBN: 978-0735619678.

Schwaber, K., & Sutherland, J. (2017). The Scrum Guide. ScrumGuides.org. Retrieved from <https://www.scrumguides.org/scrum-guide.html>

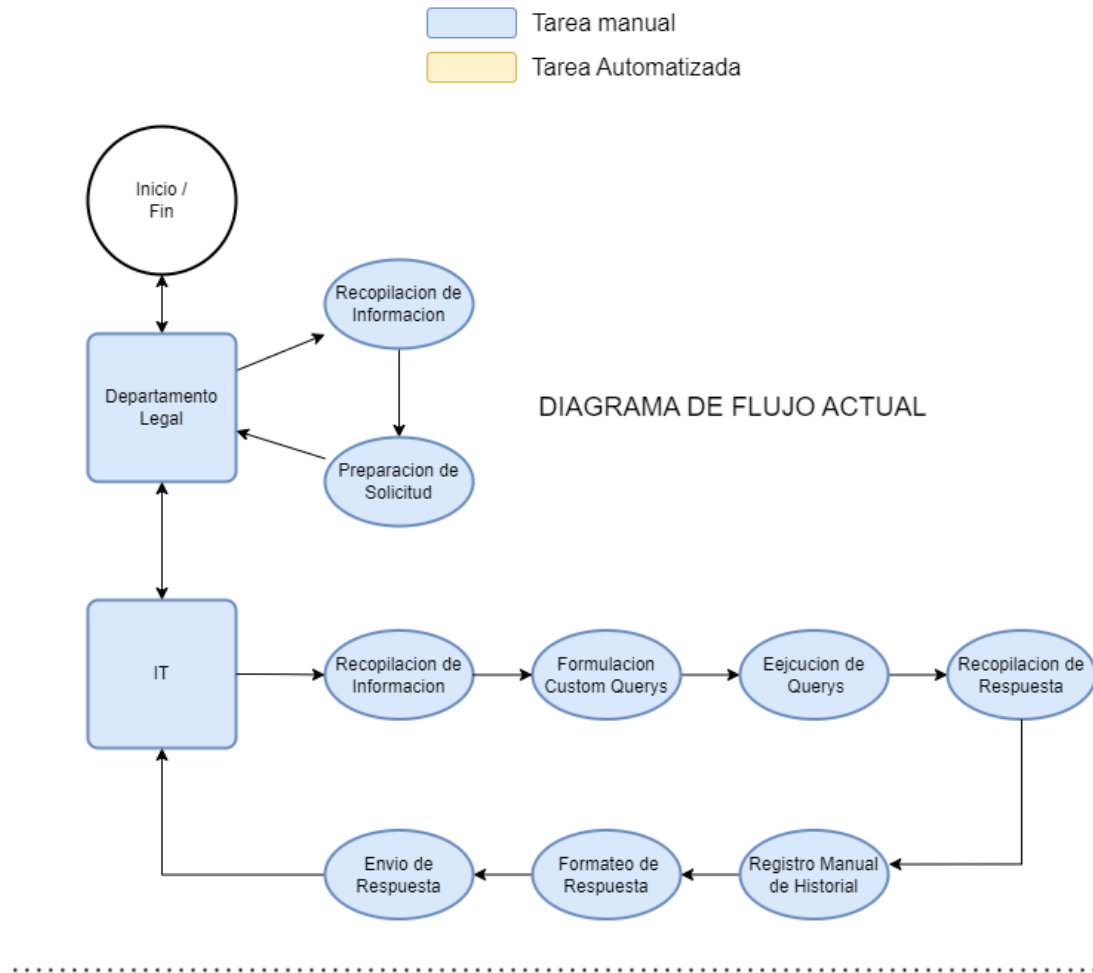
Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. Doctoral dissertation, University of California, Irvine.

Hong, Q., Kim, S., Cheung, S. C., & Czarnecki, K. (2013). A Taxonomy of JavaScript Usage for Web Page Analysis. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering (pp. 591-601).

Meszaros, G. (2007). xUnit Test Patterns: Refactoring Test Code. Addison-Wesley Professional. ISBN: 978-0131495050.

12. Anexos

Diagrama de Flujo



PROPUESTA DE AUTOMATIZACION DIAGRAMA DE FLUJO

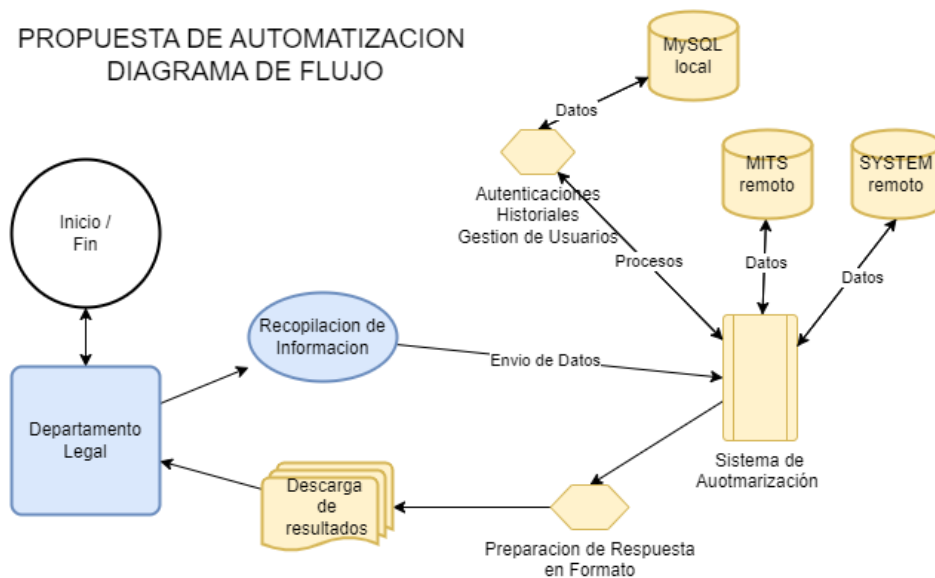
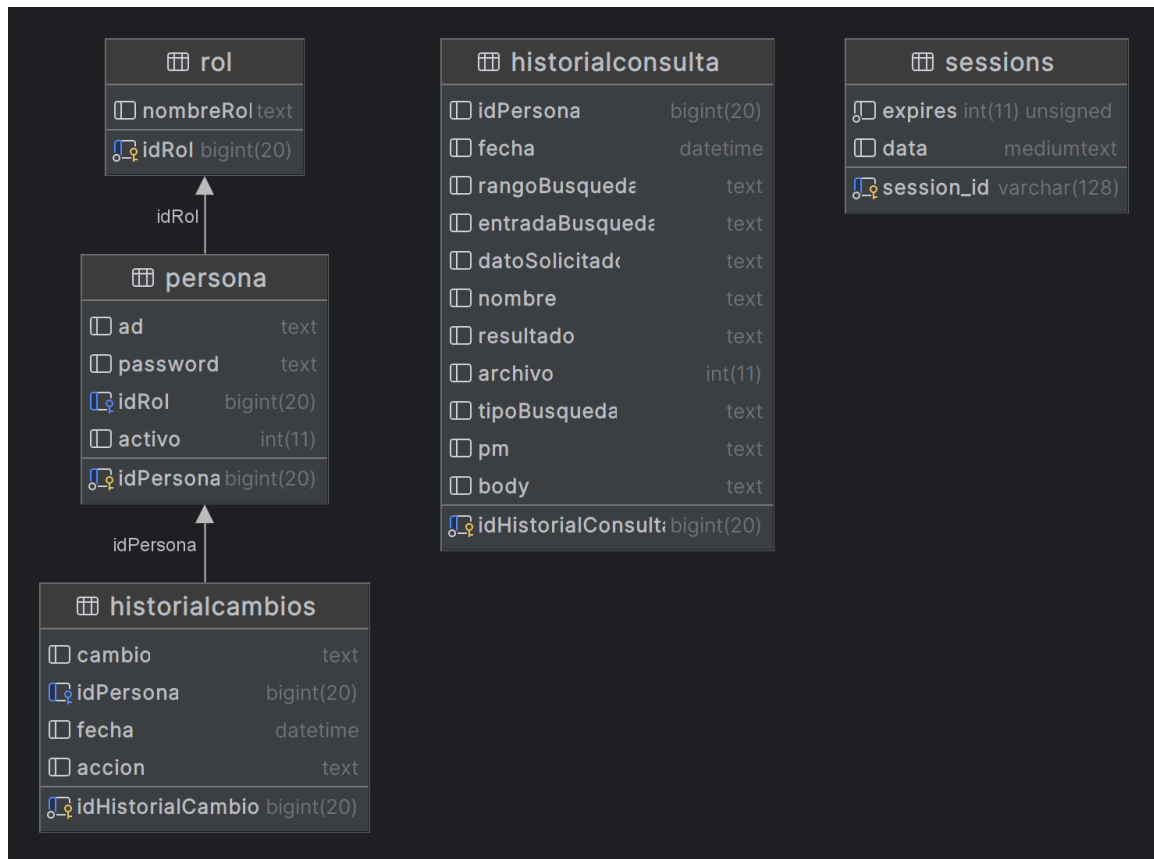


Diagrama de Base de Datos Local



Inicio de Sesión del Sistema

Formulario de Inicio de Sesión del Sistema.

Icono de usuario y candado.

Iniciar Sesión

Nombre de AD

.....

INICIAR SESIÓN

Interfaz de RRFF

Requerimientos Fiscales

Consultar un RequerimientoAdministraciónisherrera

Requerimientos Fiscales

Llenado de Datos

añadir campo

Numero de PM

☐ ULTIMOS 5 AÑOS

Fecha Inicio

Fecha Fin

☒ DATOS TITULAR

☐ IMEI TITULAR

☐ LLAMADAS

☐ SMS

☐ RADIO BASES

☐ DATOS TITULAR REF

☐ IMEI REF

☐ DETALLE RECARGAS

☐ FLUJO DE DATOS

ENVIAR

Respuesta de Requerimientos

Requerimientos Fiscales

Consultar un RequerimientoAdministraciónisherrera

Detalles de RF_XE6WvGISp

PM:

320636

Requerimiento Fiscal:

RF_XE6WvGISp

Responsable:

isherrera

Fecha de solicitud:

25 oct 2023 - 18:58:23

Datos Buscados:

72163499,60010883

Datos Solicitados:

datos titular, flujos llamadas, flujos sms, radiobases, imei

Rango Búsqueda:

2023-01-01 / 2023-10-25

Click aquí para descargar 2 archivos adjuntos

RESPUESTA REQUERIMIENTO FISCAL RF_XE6WvGISp:-60010883 (VIGENTE EN EL PERIODO SOLICITADO).-72163499 (CORRESPONDE A OTRO OPERADOR).

1. TITULAR(ES):-60010883 (NRO DE REGISTROS: 1).

TELEFONO SISTEMA TITULAR NRO_DOC FECHA_INI FECHA_FIN DIRECCION

60010883 CC POST BASE D JOSE JORGE BOZO QUISBERT 3450171 CLPZ 09/12/2011 ACTIVO D1-SCZ - ZONA LA COLORADA-BARRIO LA COLORADA CALLE A NRO 4265 # 4265, D2-SCZ - ZONA BARRIO LA COLORADA-CALLE A CUNUMISITA # 4265, D3-SCZ - ZONA BARRIO LA COLORADA-CALLA: # 4265, D4-SCZ - ZONA B/LA COLORADA-BARRIO SAN LORENZO LA COLORADA 5TO ANILLO AV/LA COLORADA # 4265, D5-SCZ - ZONA LA COLORADA- BARRIO SAN

COPIAR

Historial de Usuarios

Requerimientos Fiscales

Consultar un Requerimiento

Administración

isherrera

Historial Administrativo de isherrera

Historial RRF	PM	Nombre	Fecha Hora	Tipo	Búsqueda	Datos	Archivos	Rango de Búsqueda	Ver Resultado
Historial Administrativo									
	320743	RF_JulOLcaIT	25 oct 2023 - 20:04:42	ci	729511	datos titular, flujo llamadas	0	2023-05-15 / 2023-10-25	Ver Resultados
	320742	RF_Hlpi4DIHV	25 oct 2023 - 20:00:32	telefono	71631352	datos titular, flujo llamadas, flujos ms, radiobases	0	2023-10-20 / 2023-10-25	Ver Resultados
	320741	RF_dzv7bOLh6	25 oct 2023 - 19:57:42	ci	12424871	datos titular	0	2018-10-25 / 2023-10-25	Ver Resultados
	320669	RF_Z9roiDax	25 oct 2023 - 19:52:15	ci	13487757	datos titular	0	2018-10-25 / 2023-10-25	Ver Resultados
	320662	RF_uNA5YigEi	25 oct 2023 -	ci	10722687	datos titular	0	2018-10-25 / 2023-10-	Ver

Gestión del Sistema

Requerimientos Fiscales

 Consultar un Requerimiento

 Administración

 isherrera

Panel de Control - isherrera

Gestion de ROLES

Gestion de los Roles registrados en el Sistema.

Entrar

Crear Nuevo Usuario

Registrar nuevo usuario en el sistema.

Entrar

Ver Usuarios

Gestion de datos, busqueda, historiales.

Entrar

Control de Usuarios

Requerimientos Fiscales

 Consultar un Requerimiento

 Administración

 isherrera

Gestión de Usuarios del Sistema

AD	Rol	Activo	Acción	
jpmunoz	Administrador	Activo	Ver Historial	Modificar Perfil
fcasano	Administrador	Activo	Ver Historial	Modificar Perfil
smaldonado	Nominal	Bloqueado	Ver Historial	Modificar Perfil
marvargas	Administrador	Activo	Ver Historial	Modificar Perfil
eraranibar	Administrador	Activo	Ver Historial	Modificar Perfil
masiles	Administrador	Activo	Ver Historial	Modificar Perfil
wvargas	Administrador	Activo	Ver Historial	Modificar Perfil