

Estructura del PR

Titulo: [Nombre del backlog/tarea establecida]

[Descripción breve del desarrollo]

APIs o Servicios Afectados (cURL si es necesario)

- Listar las APIs o módulos que se crearon o modificaron y si se tiene el cURL.

Solución Implementada:

- Descripción Técnica:
 - Detallar cómo se implementaron las funcionalidades.

Pruebas automatizadas para Frontend

- Indicar la ubicación del archivo HTML del proyecto

Impacto en el Sistema:

- Componentes Afectados:
 - Módulos específicos o funcionalidades existentes que podrían verse afectadas.

Impacto en la Base de Datos:

- Descripción de nuevos campos, tablas o cambios en esquemas.
- Indicar las tablas de base de datos donde se encuentra la información.

Indispensable para el backend:

- Contrato swagger.
- Si no es un backend con servicio REST tener el contrato AsyncApi

RECORDAR:

- Documentos como local settings se mandan por privado y con link del PR.
- **El PO** debe tener el **backlog diligenciado** con **casos de uso** y con **criterios de aceptación**, además de tener una **descripción global** de lo que se quiere **desarrollar**.

- Tener en cuenta si se debe correr el desarrollo en un ambiente específico como mock, sockets, etc... mencionarlo en el PR. (De acuerdo al caso se debe dejar en el PR o mandar el archivo por privado con link del PR).

Ejemplo: Vale aclarar que para la ejecución en local del proyecto uso "ionic serve --configuration=staging" en la terminal para usar las variables de entorno establecidas en el archivo de environment.staging.ts además de ejecutar el back end del repositorio de WiscanTools en local.

- Si se añade una nueva variable de entorno o cadena de conexión BD, mencionarlo en el PR

Ejemplo: Se debe agregar al local settings la variable: "ContainerNameAlerts": "Alerts"

- Si se pide un cambio en el PR mencionarlo en los comentarios y avisar que se realizó para seguir el adecuado proceso del testeo

Ejemplo:

4 de jun

Active

Se añadió un nuevo input numérico antes del botón de envío que permite al usuario definir, de forma opcional, el tiempo en segundos entre cada batch de peticiones. Este valor es utilizado para introducir una pausa entre las solicitudes, aplicando la lógica correspondiente y mostrando las alertas informativas según el proceso realizado.

The screenshot shows the 'Portal Reader Simulator' interface. At the top, there are search filters for 'GirogUSDAS', 'Type' (set to 'Input text'), 'Id' (set to 'Id'), and 'Option: IdTransaction'. A 'Search' button is also present. The main area displays a table of transactions with columns for 'EPC' and 'Value'. Below the table, a section titled 'Data for sending' contains four input fields: a date/time field ('12:12:12 12:12:12'), a date field ('2025-06-05T02:15:23'), a numeric field ('100'), and a text field ('1'). A 'Send' button is located next to the numeric field.

Ejemplo PR: Son ejemplos de información de PRs con información completa

1.

Se realiza mejoras solicitadas al Front-end. Cabe decir que para su ejecución y pruebas en local realice las siguientes indicaciones:

- Cambio variable de archivo environment.prod.ts a localTagsApiEndpoint: '<http://localhost:7173/api>'
- Ejecución del repositorio WiscanTools en local con variable MongoDBConnectionString a la establecida para la BD de pruebas en Elite
- ionic serve --configuration=elite : Para uso de variables de producción y ejecución del proyecto
- Entrar a URL <http://localhost:8100/>

Formato Pull request:

1. APIs o Servicios Afectados:

No se modificaron APIs o servicios backend
Todos los cambios son únicamente en el frontend en la carpeta reader-simulator
Se mantienen los endpoints existentes sin alteraciones
No hay impacto en la base de datos

2. Solución Implementada:

Descripción Técnica:

- ◆ Página Implementada:
Límite por página: 100 elementos máximo por vista
Lazy Loading: Solo se renderizan los elementos de la página actual
Navegación: Controles First, Previous, Numbers, Next, Last
Estado persistente: Mantiene selecciones al navegar entre páginas
- ◆ Selección Global:
Select All Global: El checkbox principal selecciona TODOS los elementos del resultado (no solo los 100 visibles)
Estado inteligente: Distingue entre selección individual vs selección global
Persistencia: Las selecciones se mantienen al cambiar de página
Auto-detección: Sale automáticamente del modo global si se deseleccionan elementos manualmente

- ◆ Contador Dinámico:
Actualización en tiempo real: Total selected: X se actualiza con cada clic
Getter computed: Calcula automáticamente el total real de elementos seleccionados
Indicadores visuales: Muestra información de paginación: "1-100 of 500 items"
- ◆ Funcionalidades Técnicas Agregadas:
ITEMS_PER_PAGE = 100 - Constante de paginación
allData[] - Array completo de resultados
allSelectedElements: Map<string, any> - Estado global de selecciones
isSelectAllMode: boolean - Trackea modo de selección global
totalSelectedCount getter - Contador dinámico reactivo
Métodos de navegación: goToPage(), getPaginatedData(), updateCurrentPageData()

3. Pruebas Realizadas:

Se realizan pruebas manuales no documentadas.
 Pruebas adicionales pendientes - A realizar

4. Impacto en el Sistema:

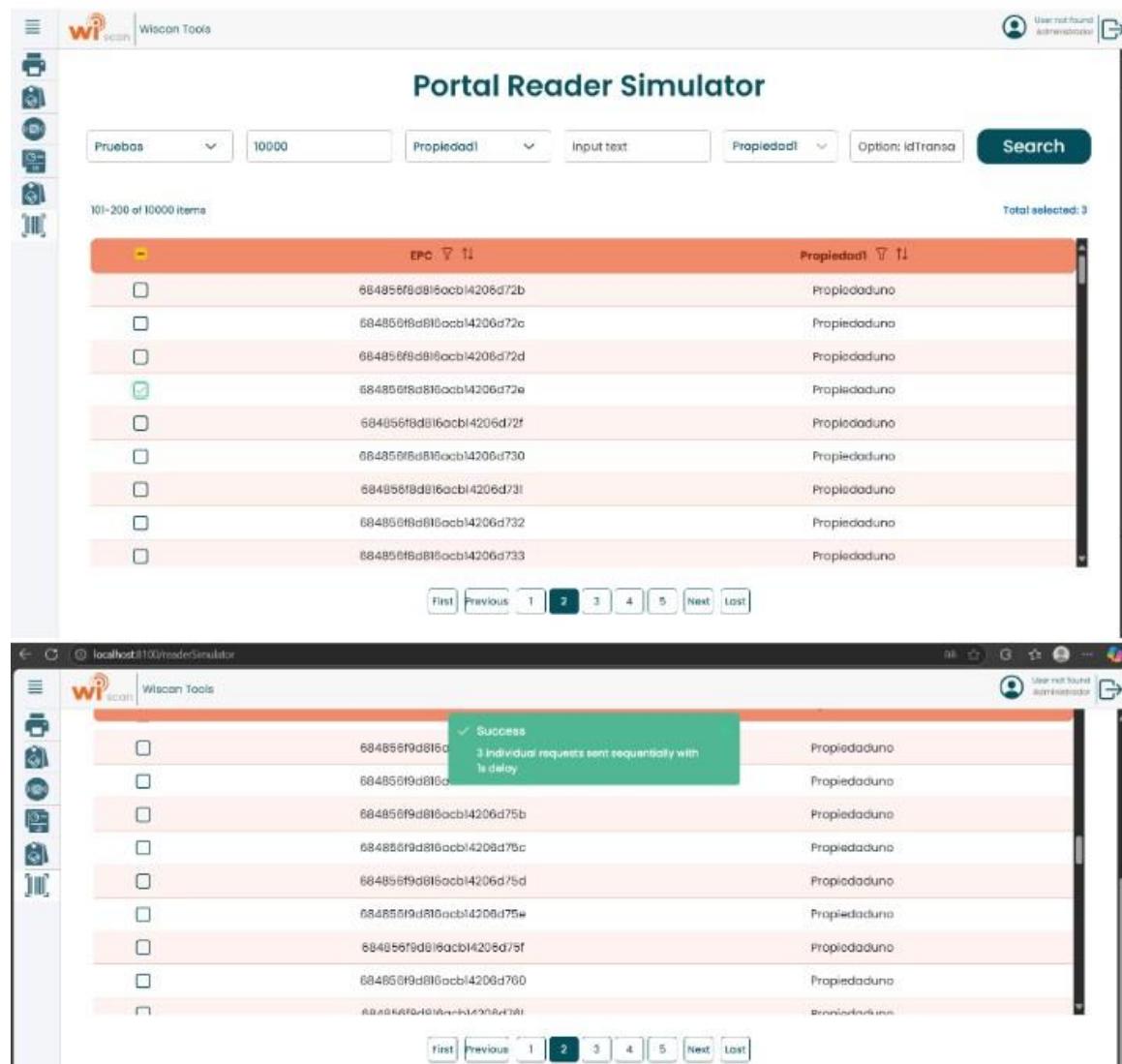
Componentes Afectados:

◆ Afectados Directamente:

reader-simulator.component - Único módulo modificado

Funcionalidad de filtrado - Ahora incluye paginación automática

Lógica de envío - Mantiene compatibilidad con elementos seleccionados globalmente



The screenshot shows a web-based application titled "Portal Reader Simulator". The interface includes a sidebar with various icons, a top navigation bar with "Wiscon Tools" and user authentication information, and a main content area with search and filter options. The main table displays 101-200 of 10000 items, with columns for "EPC" and "Propiedad". A modal or tooltip is overlaid on the third item from the top, containing the text: "Success: 3 individual requests sent sequentially with 1s delay".

EPC	Propiedad
684856f8d816acb14206d72b	Propiedaduno
684856f8d816acb14206d72c	Propiedaduno
684856f8d816acb14206d72d	Propiedaduno
684856f8d816acb14206d72e	Propiedaduno
684856f8d816acb14206d72f	Propiedaduno
684856f8d816acb14206d730	Propiedaduno
684856f8d816acb14206d731	Propiedaduno
684856f8d816acb14206d732	Propiedaduno
684856f8d816acb14206d733	Propiedaduno

2.

💡 Resumen del Pull Request

Se implementaron nuevas funcionalidades de filtrado, mejoras en validación y preservación de inputs, y se unificó la interfaz visual del componente reader-simulator con diseño actual establecido. Además, se realizaron ajustes técnicos y correcciones de errores detectados en el flujo de simulación de etiquetas RFID.

☒ Cambios realizados en:

localTagsApiEndpoint:

Se eliminó la porción /Tags del endpoint en el archivo environment.ts para los entornos local y staging.

Componente reader-simulator.

⚙️ Descripción técnica:

Nuevo input de exclusión por IdTransaction:

Se agregó un campo opcional para excluir múltiples IDs de transacción del filtrado (formato: 32,45,67).

Lógica de filtrado mejorada:

Cuando no se define ninguna propiedad específica, ahora se filtran las etiquetas por tipo en lugar de realizar la petición de esa propiedad con un valor vacío.

Validación de inputs numéricos:

Se preserva el último valor válido ante entradas incorrectas. El valor del campo limit se conserva correctamente al cambiar de plantilla.

UI/UX:

Se actualizó el esquema de colores utilizando --ion-color-primary-p500 para unificar visualmente con el módulo properties-manager.

También se estandarizaron los estilos de títulos, botones y formularios.

Gestión del ciclo de vida del componente:

Mejor manejo del estado de los inputs y su reinicio, especialmente en el campo de límite de resultados.

Interacción del botón de búsqueda:

Se deshabilita el botón "Search" y cambia su texto a "Query..." mientras se obtiene la respuesta del backend.

Alerta de repetición de búsqueda:

Se añade alerta al intentar ejecutar la misma consulta con parámetros idénticos a los de la solicitud anterior.

☒ Gestión de ramas:

Esta rama fue creada a partir de integraciónApiHistorico y se realizó merge con feat/rfidTagsReaderSimulator sin conflictos.

💡 Notas de testing:

Se dejaron console.log de la petición y respuesta para facilitar pruebas por parte del área de Testing.

Pendiente la implementación de pruebas unitarias.

The screenshot shows the 'Portal Reader Simulator' interface. On the left, there's a sidebar with icons for TestPortal, MAC Address, Date, Input text, Category, and Options. The main area has a header 'Portal Reader Simulator' with tabs for 'TestPortal' (selected), 'MAC Address', '2020-09-30T23:59:59Z', 'Input text', 'Category', and 'Options: IdTransaction'. Below the header is a table with columns 'EPC', 'V', 'TL' and 'Category'. The table lists several tags with their EPC numbers and categories. A specific row is highlighted in pink, showing '9980216A1F5B98023E123097' under 'EPC', 'None' under 'V', 'None' under 'TL', and 'None' under 'Category'. At the bottom, there's a section for 'Data for sending' with fields for 'MAC Address (XX:XX:XX)', '2020-09-30T23:59:59Z', '100', and a 'Send' button. To the right of the table, a sidebar shows the JSON structure of the selected tag: { "EPC": "9980216A1F5B98023E123097", "V": null, "TL": null, "Category": null }. The JSON tree also includes 'Request' and 'Response' sections.

Estructura para Bugs (Planeados)

Título: [Nombre del bug]

[Descripción breve del problema y la solución]

Causa del Bug:

- Descripción breve de lo que sucedía.

Impacto:

- Mencionar qué partes del sistema se ven afectadas.

Solución Implementada:

- Explicación clara de los cambios realizados
- APIs afectadas o módulos modificados

Evidencia de la Solución:

- Capturas de pantalla:
 - Después de la corrección, mostrando el comportamiento esperado y el comportamiento que se presentaba.
- Videos:
 - Ejecución completa que demuestre la corrección.
- Logs actualizados:
 - Verificación de que ya no aparecen errores tras el fix.

RECORDAR

- Cuando el bug se presenta durante el sprint y se sube el PR el backlog donde pertenece esa tarea debe estar diligenciado en el momento de hacer el testing, de lo contrario se quedará en espera.
- El PO debe diligenciar el backlog con capturas, videos o cualquier elemento que haga evidencia del bug y se vea claramente el error.
- El desarrollador en lo posible debe replicar el bug y dejarlo en evidencia.

Ejemplo:

Link de acceso:

<http://localhost:8100/printingServices> ↗

El problema era que no se estaba mostrando las etiquetas cuando había una agrupación y se quería hacer una segunda consulta después de la primera.

[Grabación de pantalla 2025-01-20 a la\(s\) 10.36.30 a.m..mov](#)

Versión con el cambio:

[Wiscan - Google Chrome 2025-01-20 10-40-44.mp4](#)

- Tener en cuenta que, si se debe correr en un entorno específico, si se realizó cambios en el local settings o en el proyecto, se debe mencionar en el PR.