

# Yggdrasil Bounty Record

## Lista de Sharepoint conectada a base de datos en SQL Server

Leoncio D. Gómez A.

**Resumen**—El presente informe expone el proceso de implementación de un sistema constituido por una lista creada en Sharepoint cuyos cambios serán reflejados en una base de datos creada en SQL Server; en este sentido, se desarrollarán dos diferentes aproximaciones, la primera consistirá en un WebPart a ser desplegado directamente en Sharepoint que; a través de un backend y un frontend, llevará a cabo las distintas operaciones con los registros tanto en la lista como en la tabla; posteriormente, se expondrán las acciones realizadas para el establecimiento de la conexión directa entre la referida lista de Sharepoint con la base de datos a través de diferentes flujos creados por medio de Power Automate.

This report exposes the implementation process of a system made up of a list created in Sharepoint whose changes will be reflected in a database created in SQL Server; In this sense, two different approaches will be developed, the first will consist of a WebPart to be deployed directly in Sharepoint that; through a backend and a frontend, it will carry out the different operations with the records both in the list and in the table; Subsequently, the actions carried out to establish the direct connection between the aforementioned Sharepoint list with the database through different flows created through Power Automate will be presented.

**Index Terms**—Yuxi, Sharepoint, SQL Server, IEEE, journal,  $\LaTeX$ , paper, template.

### 1. INTRODUCCIÓN

Estamos creando una herramienta interna dentro de Yuxi y realmente queremos usar Sharepoint para facilitar complementos adicionales. Al igual que las integraciones de Power Automate, necesitamos conectar Sharepoint con el servidor SQL a través del sitio web de listas, la base de datos se proporcionará como parte de la investigación y todos deben crear su propio esquema/tabla y realizar la conexión

#### 1.1. SharePoint WebPart

De acuerdo a [1], los WebPart de Sharepoint son controles del lado del servidor que se ejecutan dentro de una página de elementos web. Existen herramientas que permiten obtener rápidamente un andamiaje (*Scaffolding*) para construir este tipo de componentes utilizando librerías como React, el cual fue el recurso escogido para tal fin. Una explicación detallada de como crear este andamiaje empleando Visual Studio Code, puede ser consultada en el sitio web de [Microsoft Learn](https://learn.microsoft.com/).

Una vez obtenido el andamiaje del WebPart, se implementa el modelo ilustrado en la figura 1

##### 1.1.1. Servicio SQL Server

Consiste en una clase del WebPart implementado con la finalidad de agrupar todas las definiciones de los endpoints que serán llamados con la finalidad de crear, obtener, actualizar y borrar registros desde la UI del WebPart hasta

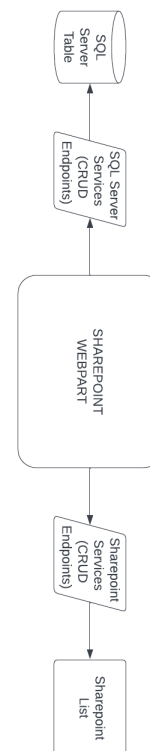
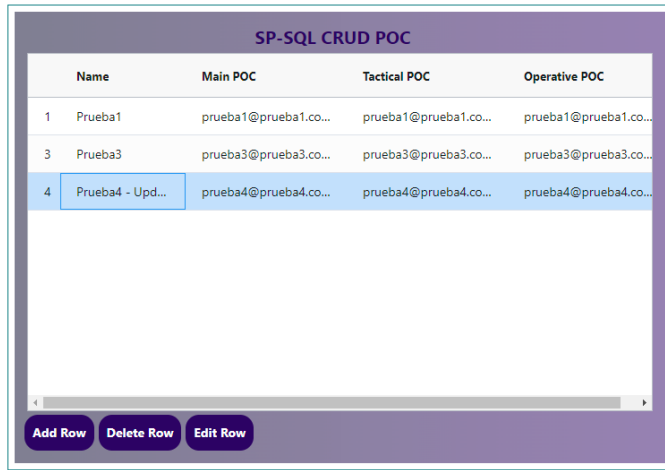


Figura 1. Modelo del WebPart Sharepoint

- Desarrollador de Software FullStack, Yuxi Global. Medellín, Colombia  
Ingeniero de Telecomunicaciones, Universidad de Carabobo. Bárbula, Venezuela  
E-mail: leoncio.gomez@yuxiglobal.com



	Name	Main POC	Tactical POC	Operative POC
1	Prueba1	prueba1@prueba1.co...	prueba1@prueba1.co...	prueba1@prueba1.co...
3	Prueba3	prueba3@prueba3.co...	prueba3@prueba3.co...	prueba3@prueba3.co...
4	Prueba4 - Upd...	prueba4@prueba4.co...	prueba4@prueba4.co...	prueba4@prueba4.co...

Figura 2. Webpart: Lista de items contenidos en la base de datos

la base de datos de SQL Server; a través de un WebAPI implementado para tal fin.

A continuación se mencionan los referidos endpoints:

- **Create:** *baseUrl/Api/Client/InsertClientDetails*
- **Retrieve:** *baseUrl/Api/Client/GetClientsDetails*
- **Update:** *baseUrl/Api/Client/UpdateClientDetails*
- **Delete:** *baseUrl/Api/Client/DeleteClientDetails*

### 1.1.2. Servicio Sharepoint

Constituye el homólogo del servicio SQL Server, encargado de efectuar las operaciones CRUD hacia la lista de Sharepoint, esta vez llamando al [servicio REST](#) proporcionado por esta herramienta.

En la implementación de esta solución, se emplearon los siguientes endpoints, considerando que el valor del baseUrl está relacionado con la cuenta Sharpoint y el sitio donde está alojada la lista a la cual se desea vincular con la base de datos:

**BaseURL:** `https://lncgomz.sharepoint.com/sites/lncgomzdeveloper/_api/Web/Lists/getbytitle('SP_SQLSERVER')`

Donde *SP\_SQLSERVER* es el nombre de la [lista](#) en cuestión.

- **Create:** *baseUrl/Items* (POST)
- **Retrieve:** *baseUrl/Items* (GET)
- **Update:** *baseUrl/Items/getbyid(: id)* (POST)
- **Delete:** *baseUrl/Items/getbyid(: id)* (DELETE)

### 1.1.3. Interfaz de Usuario

En la figura 2 puede apreciarse la pantalla principal del Webpart implementado a partir del andamiaje proporcionado por el script especializado.

Del mismo modo, se ha diseñado una ventana compartida para las operaciones de adición y edición de items.<sup>3</sup>

## 1.2. Power Automate

El inconveniente en la aproximación anterior consiste en el hecho de no poder realizar la edición de entradas

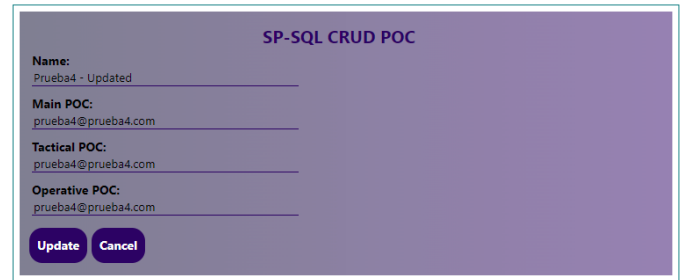


Figura 3. WebPart: Adición y Edición de Items

directamente en la lista de sharepoint; sino a través de la interfaz de usuario del webpart, lo cual no permite aprovechar las otras funcionalidades incorporadas en este elemento de Sharepoint; razón por la cual se hizo necesario explorar otras aproximaciones para la implementación del sistema solicitado; en ese sentido, se pudo determinar que a través de la herramienta Power Automate, podía establecerse una conexión directa entre la lista y la base de datos de SQL Server, sin la necesidad de desarrollar un backend o un frontend adicionales.

Según la documentación oficial [2] Power Automate es un servicio que ayuda a crear flujos de trabajo automatizados entre aplicaciones y servicios para sincronizar archivos, recibir notificaciones, recopilar datos y más.

En términos prácticos, funciona de forma similar a las macros de Office pero su diseño se lleva a cabo de manera más intuitiva, cuenta con numerosas plantillas para tareas frecuentes y permite la conexión entre una gran variedad de aplicaciones y servicios; por ejemplo, SharePoint y SQLServer.

Una característica resaltante, en el diseño de estos flujos automatizados, es que el proceso sigue el paradigma *Codeless* lo cual implica que se requiere ningún o muy poco código para implementarlos; en su lugar, basta con identificar un disparador (*Trigger*) y la correspondiente acción (*Action*) para cada uno de estos parámetros, es necesario indicar unos pocos valores de configuración; como por ejemplo, las credenciales de conexión a base de datos, la URL del sitio donde se encuentra la lista de sharepoint que queremos conectar, etc. Sin embargo, es posible aplicar un procesamiento a los datos a medida que avanzan en el flujo; por otro lado, una desventaja de las soluciones de Power Automate, consiste en el hecho de tener que pagar una licencia adicional por los derechos de uso de esta herramienta <sup>4</sup>.

### 1.2.1. Evento 'Fila Agregada'

El primer flujo implementado consiste en capturar el evento de agregar una nueva fila a la lista de sharepoint para ejecutar automáticamente un INSERT en la base de datos SQL Server con la misma información agregada en la mencionada lista <sup>5</sup>

### 1.2.2. Evento 'Fila Actualizada'

Posteriormente, se implementa el flujo que captura el evento de editar una fila existente en la lista de sharepoint para ejecutar automáticamente un UPDATE en la base de datos SQL Server, usando como referencia el Id de la fila en cuestión. <sup>6</sup>

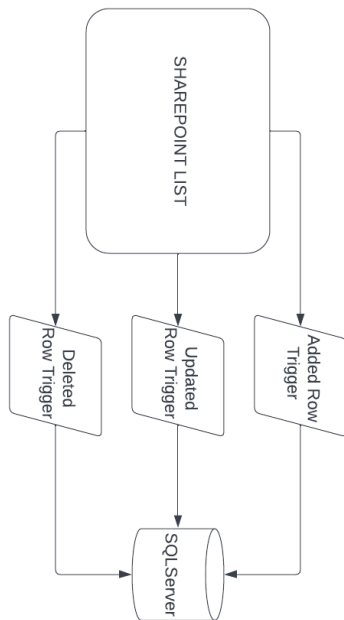


Figura 4. Modelo de Power Automate Flows

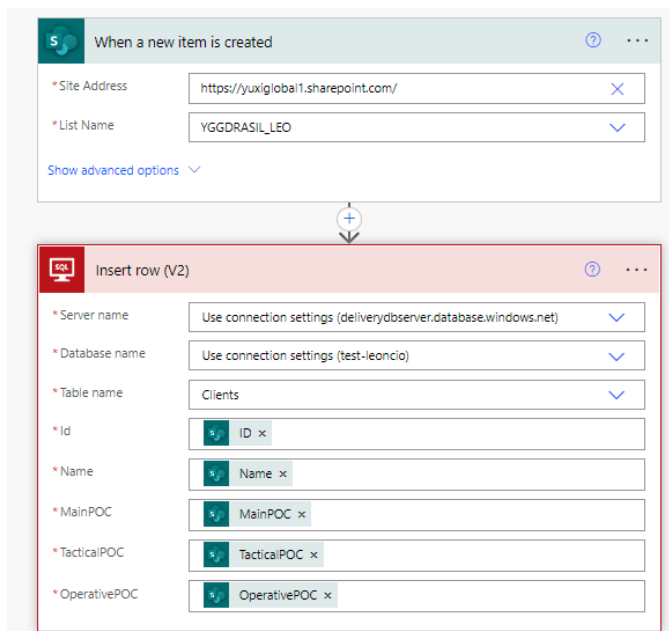


Figura 5. Flujo 'Fila Agregada'

### 1.2.3. Evento 'Fila Eliminada'

Por último, a partir de la eliminación de una fila en la lista de sharepoint y tomando como parámetro de referencia el Id de la entrada eliminada, se automatiza la ejecución de un DELETE donde el Id de la entrada en la base de datos, coincida con la del registro eliminado. <sup>7</sup>

Además de poder implementar las funcionalidades requeridas de una manera rápida e intuitiva, Power Automate ofrece un dashboard donde puede monitorearse todas las ejecuciones de cada uno de los disparadores, así como un informe detallados en caso de haber una ejecución fallida, lo cual constituyen ventajas importantes con respecto a la

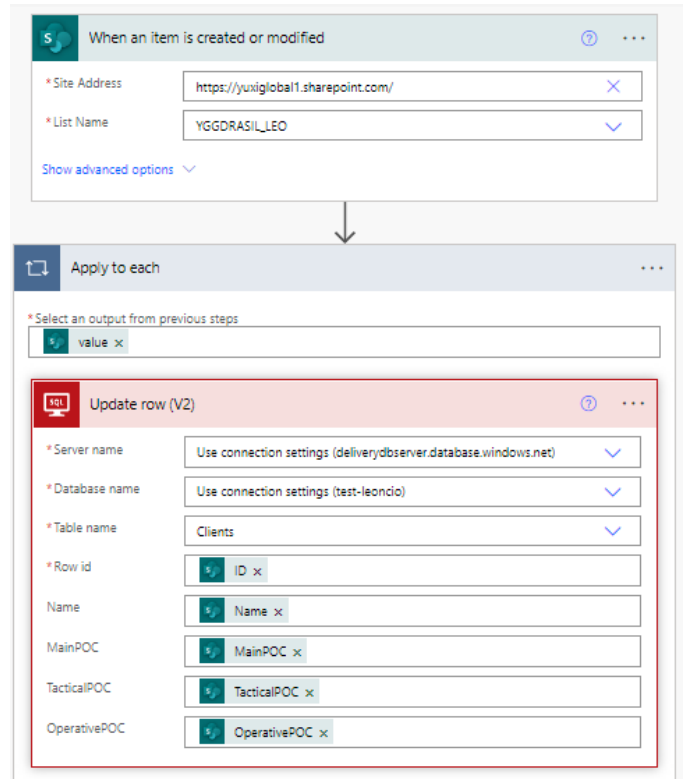


Figura 6. Flujo 'Fila Actualizada'

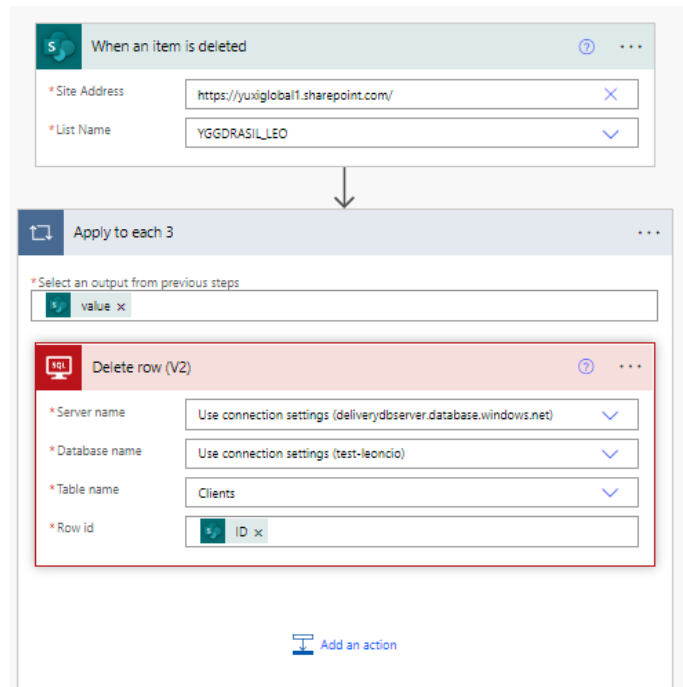


Figura 7. Flujo 'Fila Eliminada'

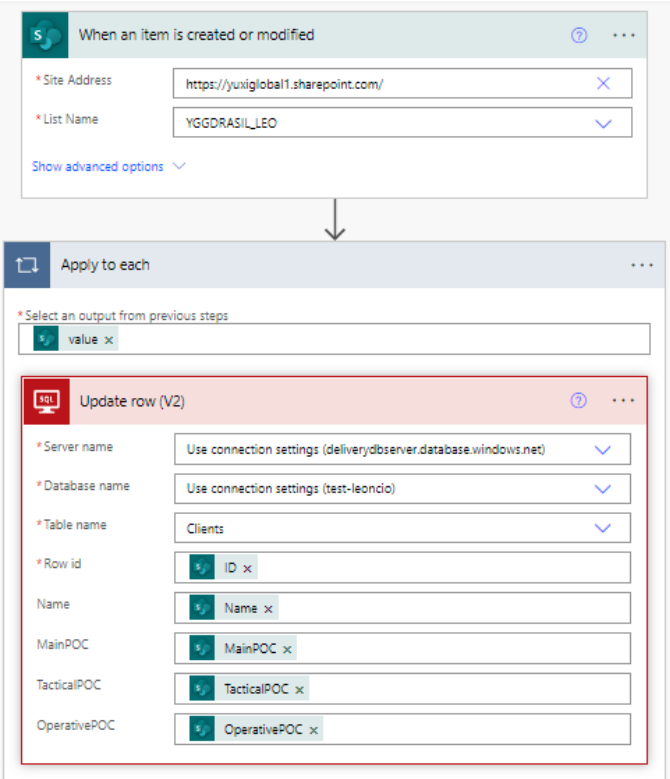


Figura 8. Flujo 'Fila Actualizada'

primera aproximación propuesta.

2. CONCLUSION

El componente solicitado pudo ser implementado a través de dos aproximaciones distintas; a saber, desarrollando un WebPart en Sharepoint y por medio de los flujos de Power Automate. Ambas alternativas cuentan con ventajas y desventajas; en el primer caso, no se requiere el pago de licencias adicionales, se dispone de un control total de todo el flujo de información y permite extender el comportamiento del sistema al desarrollar funciones adicionales ya que es una solución de código; sin embargo, esto también significa que el proyecto requiere el desarrollo y despliegue de dos componentes, cómo mínimo (Backend y Frontend) y ambos deben encontrarse operativos para que todo el sistema funcione. En el otro lado, la alternativa que hace uso de los flujos de Power Automate no requiere del uso de componentes adicionales, integra diferentes aplicaciones, servicios y acciones personalizables cuya implementación no requiere de codificación, ni despliegues adicionales; no obstante, el aprovechamiento de estas herramientas requiere del pago de licencias cuyos costo; sin embargo, debe sopesarse con un menor tiempo de desarrollo, fácil escalabilidad y el costo de mantenimiento y almacenamiento del webapi requerido en la otra alternativa.

ÍNDICE

1.	Introducción	1
1.1.	SharePoint WebPart . . . . .	1
1.1.1.	Servicio SQL Server . . . . .	1
1.1.2.	Servicio Sharepoint . . . . .	2
1.1.3.	Interfaz de Usuario . . . . .	2
1.2.	Power Automate . . . . .	2
1.2.1.	Evento 'Fila Agregada' . . . . .	2
1.2.2.	Evento 'Fila Actualizada' . . . . .	2
1.2.3.	Evento 'Fila Eliminada' . . . . .	3
2.	Conclusion	4
	Referencias	4

REFERENCIAS

[1] Microsoft Learn, Create web parts for SharePoint, Online <https://learn.microsoft.com/en-us/visualstudio/sharepoint/creating-web-parts-for-sharepoint?view=vs-2022> November 3rd, 2022.

[2] Microsoft Learn, Microsoft Power Automate documentation, Online <https://learn.microsoft.com/en-us/power-automate/> November 3rd, 2022.