

Yggdrasil Bounty Record Sharepoint list connected to SQL Server database

Leoncio D. Gómez A.

Abstract—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, L^AT_EX, paper, template.

1 INTRODUCTION

WE are building an internal tool within Yuxi and really want to use Sharepoint to facilitate additional plugins. Like the Power Automate integrations, we need to connect Sharepoint to the SQL server via the Lists website, the database will be provided as part of the investigation, and everyone needs to create their own schema/table and make the connection.

1.1 SharePoint WebPart

According to [1], Sharepoint WebParts are server-side controls that run inside a web part page. There are tools that allow the developers to quickly obtain a scaffolding to build this type of components using libraries such as React, which was the resource chosen for this purpose. A detailed explanation of how to create this scaffolding using Visual Studio Code can be found on the [Microsoft Learn](#) website. Once the WebPart scaffolding has been obtained, the model illustrated in figure 1 is implemented.

1.1.1 SQL Server Service

It consists of a WebPart class implemented in order to group all the definitions of the endpoints that will be called in order to create, obtain, update and delete records from the WebPart UI to the SQL Server database; through a WebAPI implemented for this purpose.

The aforementioned endpoints are listed below:

- **Create:** *baseUrl/Api/Client/InsertClientDetails*
 - **Retrieve:** *baseUrl/Api/Client/GetClientsDetails*
 - **Update:** *baseUrl/Api/Client/UpdateClientDetails*
 - **Delete:** *baseUrl/Api/Client/DeleteClientDetails*
-
- *Desarrollador de FullStack Software Developer, Yuxi Global. Medellín, Colombia*
Telecommunications Engineer, Universidad de Carabobo. Bárbula, Venezuela
E-mail: leoncio.gomez@yuxiglobal.com



Fig. 1. WebPart Sharepoint Model

1.1.2 Sharepoint Service

It is the counterpart of the SQL Server service, in charge of carrying out CRUD operations towards the Sharepoint list, this time by calling **REST-service** provided by this tool. In the implementation of this solution, the following endpoints were used, considering that the value of the baseUrl is related to the Sharepoint account and the site where the list to which the developers want to link to the database is

	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...

Fig. 2. Webpart: List of items contained in the database

SP-SQL CRUD POC

Name:

Main POC:

Tactical POC:

Operative POC:

Fig. 3. WebPart: Adding and Editing items

hosted:

BaseURL: [https://lncgomz.sharepoint.com/sites/lncgomzdeveloper/_api/Web/Lists/getbytitle\('SP_SQLSERVER'\)](https://lncgomz.sharepoint.com/sites/lncgomzdeveloper/_api/Web/Lists/getbytitle('SP_SQLSERVER'))

Where *SP_SQLSERVER* is the sharepoint list's name.

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

1.1.3 User Interface

Figure 2 shows the main screen of the Webpart implemented from the scaffolding provided by the specialized script.

In the same way, a shared window has been designed for the operations of adding and editing items.³

1.2 Power Automate

The inconvenience in the previous approach consists in the fact of not being able to edit entries directly in the sharepoint list; but through the user interface of the web part, which does not allow taking advantage of the other functionalities incorporated in this element of Sharepoint; reason why it became necessary to explore other approaches for the implementation of the requested system; In this sense, it was determined that through the Power Automate tool, a direct connection could be established between the list and the SQL Server database, without the need to develop an

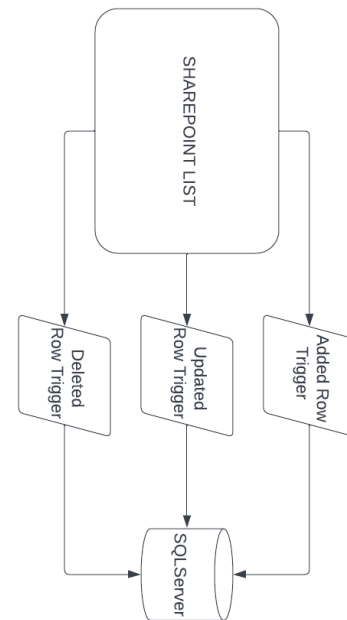


Fig. 4. Power Automate Flows Model

additional backend or frontend.

According to the official documentation [2] Power Automate is a service that helps create automated workflows between applications and services to sync files, receive notifications, collect data, and more.

In practical terms, it works in a similar way to Office macros but its design is carried out in a more intuitive way, it has numerous templates for frequent tasks and it allows the connection between a wide variety of applications and services; for example, SharePoint and SQLServer.

An outstanding feature in the design of these automated flows is that the process follows the *Codeless* paradigm, which implies that no or very little code is required to implement them; instead, it is enough to identify a trigger and the corresponding action for each of these parameters, it is necessary to indicate a few configuration values; For example, the database connection credentials, the URL of the site where the sharepoint list we want to connect to is located, etc. However, it is possible to apply processing to the data as it moves through the stream; On the other hand, a disadvantage of the Power Automate solutions consists in the fact of having to pay an additional license for the rights to use this tool ⁴.

1.2.1 'Added Row' Event

The first flow implemented consists of capturing the event of adding a new row to the sharepoint list to automatically execute an INSERT in the SQL Server database with the same information added to the aforementioned list ⁵.

1.2.2 'Updated Row' Event

Subsequently, the flow that captures the event of editing an existing row in the sharepoint list is implemented to automatically execute an UPDATE in the SQL Server database, using the Id of the row in question as a reference. ⁶

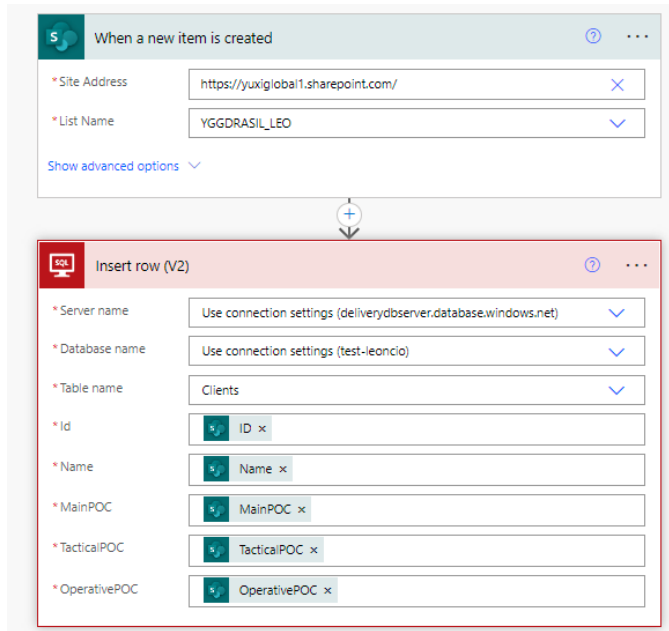


Fig. 5. 'Added Row' flow

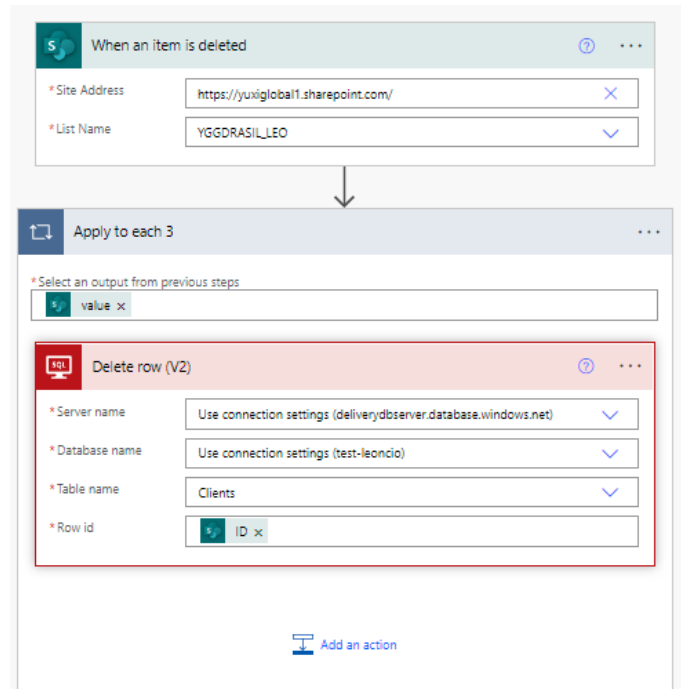


Fig. 7. 'Deleted Row' flow

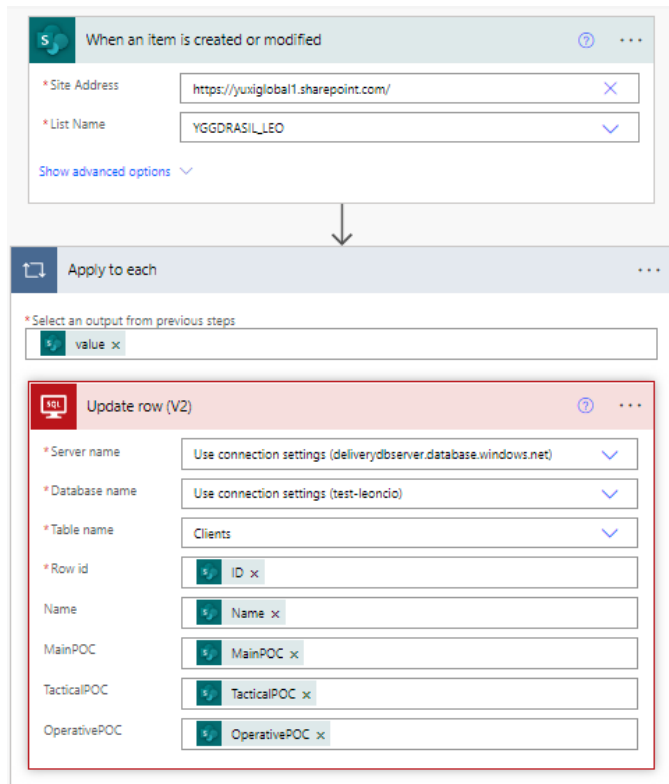


Fig. 6. 'Updated Row' flow

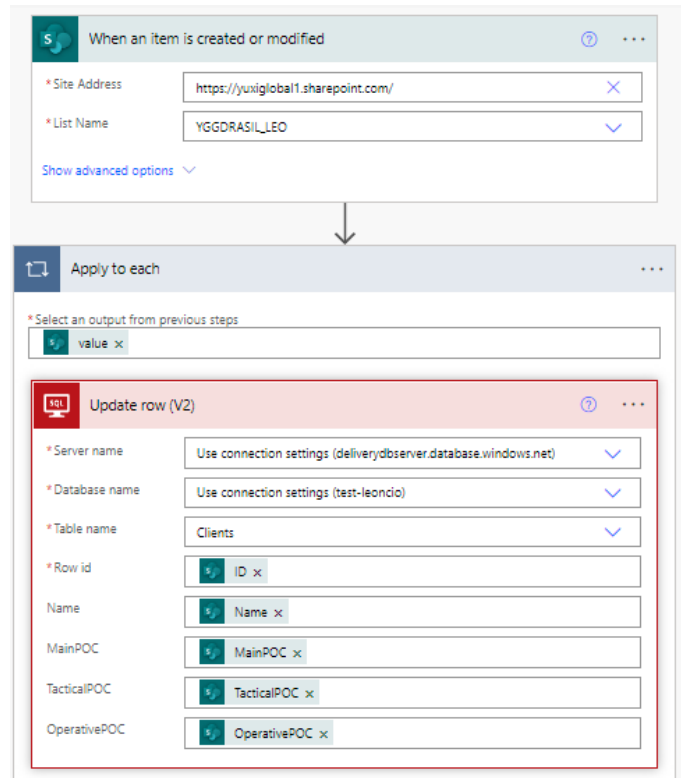


Fig. 8. 'Updated Row' flow

1.2.3 'Deleted Row' Event

Finally, after deleting a row in the sharepoint list and taking the Id of the deleted entry as a reference parameter, the execution of a DELETE is automated where the Id of the entry in the database matches that of the deleted record.⁷

In addition to being able to implement the required func-

tionalties quickly and intuitively, Power Automate offers a dashboard where all the executions of each of the triggers can be monitored, as well as a detailed report in the event of a failed execution, which are important advantages. with respect to the first proposed approximation.

2 CONCLUSION

The requested component can be implemented through two different approaches; namely, developing a WebPart in Sharepoint and through Power Automate flows. Both alternatives have advantages and disadvantages; in the first case, the payment of additional licenses is not required, there is total control of the entire flow of information and it allows the behavior of the system to be extended by developing additional functions since it is a code solution; however, this also means that the project requires the development and deployment of at least two components (Backend and Frontend) and both must be operational for the entire system to work. On the other hand, the alternative that makes use of Power Automate flows does not require the use of additional components, it integrates different applications, services and customizable actions whose implementation does not require coding or additional deployments; however, the use of these tools requires the payment of licenses whose cost; however, it must be weighed against a shorter development time, easy scalability, and the cost of maintaining and storing the webapi required in the other alternative.

CONTENTS

1 Introduction 1

1.1 SharePoint WebPart 1

1.1.1 SQL Server Service 1

1.1.2 Sharepoint Service 1

1.1.3 User Interface 2

1.2 Power Automate 2

1.2.1 'Added Row' Event 2

1.2.2 'Updated Row' Event 2

1.2.3 'Deleted Row' Event 3

2 Conclusion 4

References 4

REFERENCES

[1] Microsoft Learn, *Create web parts for Share-Point*, 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.