# Chapter 14: Queries and Reports with the Data Service Module

Welcome to this chapter dedicated to the BIZUIT Data Services module. Throughout it, we will learn to consult and process information from SQL queries or data generated by native processes of the platform, to then transform them into clear, dynamic and visually attractive reports.

We will see how to configure parameters that allow you to customize reports in real time, design interactive graphs that facilitate the interpretation of data and create connections between reports through the drilldown functionality, achieving a fluid and contextual navigation with a single click.

By the end of this class, we will have acquired the tools to:

- Build optimized SQL queries to efficiently extract and structure data.
- Configure parameters that adapt the display according to the user's needs.
- Design clear, dynamic graphics that are consistent with the objectives of the report.
- Implement *drilldowns* to explore related information in an agile and intuitive way.

No matter our level of previous experience: the approach will be practical and progressive, ensuring that we master every step, from the basics to advanced features.

Let's start this journey towards creating reports that transform data into decisions!

### Ideal Audience

This chapter is aimed at professionals who want to create advanced queries and generate graphical reports in BIZUIT, and who need to integrate data from external sources. Basic knowledge of relational databases, SQL syntax, and data visualization is recommended.

### Objectives

1. **Create SQL queries:** Design queries that extract, organize, and filter data based on specific needs.
2. **Configure dynamic parameters:** Customize reports by defining filters and default values.
3. **Design custom charts:** Generate clear and effective visualizations to analyze data.
4. **Implement drilldowns:** Create interactive connections between reports that facilitate data navigation and exploration.

# Unit 1: Introduction to the "Data Services" Module

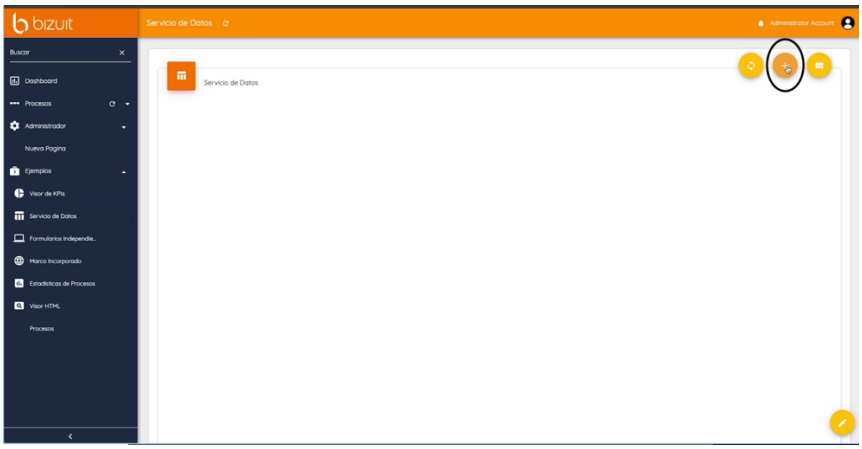
In this unit, we will delve into the BIZUIT Data Services module, a tool designed to organize, analyze, and visualize information in a centralized and efficient way. Throughout the session we will learn how to create dynamic reports, configure global filters, and establish connections between different sections of data to build a clear, interactive analysis environment adapted to the needs of the business.

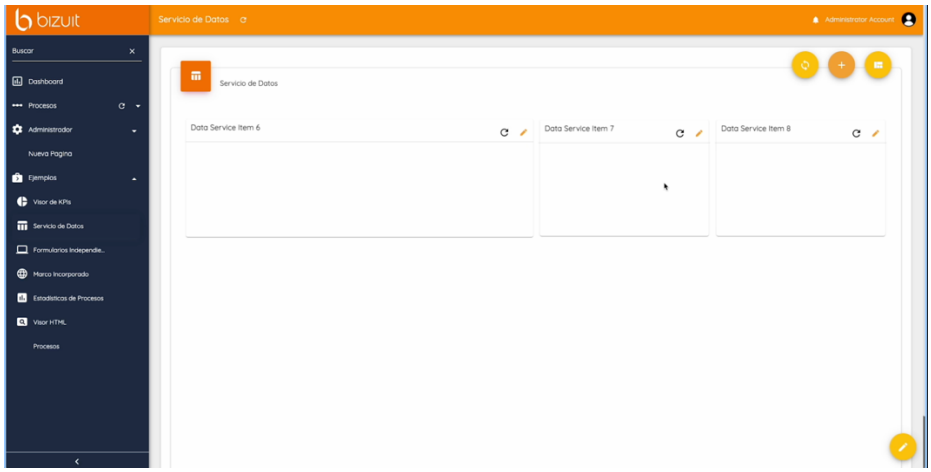
We'll start by incorporating the Data Service module into a one-page section of the Dashboard, and then move on to creating and organizing submodules, configuring global parameters, and optimizing their visual layout.

## Submodules and Global Parameters

### Submodules

We can think of submodules as independent blocks of information within the module. Each one can serve a specific purpose: to show tables, graphs or specific reports. Personalization is key, as we can define the space they will occupy and their arrangement within the page.
To add a submodule, simply select the Add option, define the percentage of the screen it will occupy and, if necessary, rearrange it later through the Submodule Layout mode, adjusting position and size to fit the rest of the elements. If at any time we want to delete it, just select the corresponding icon.






## Global Parameters

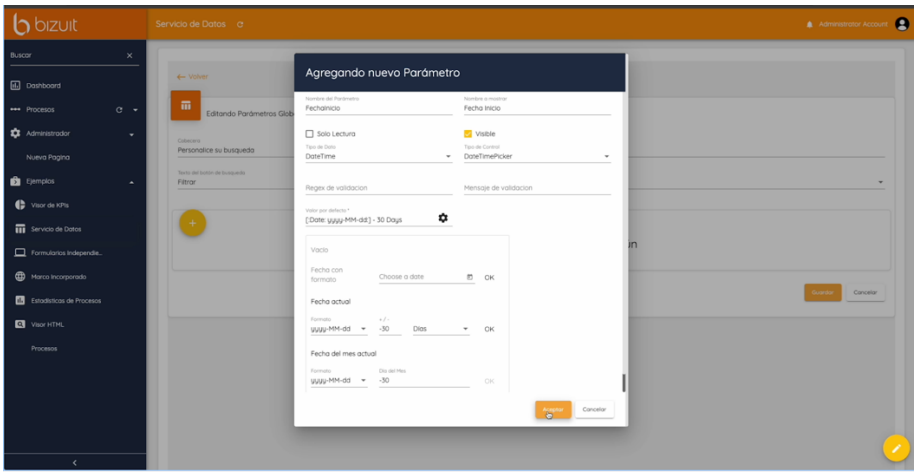
Global parameters function as filters shared by all submodules of the module. Instead of setting a filter on each one separately, we can define a set of central parameters that unify the information displayed.

For example, to filter all reports on a page by a date range, we can create the Start Date and EndDate parameters. This way, all submodules of the module will automatically use that same filter, ensuring consistency and reducing configuration work.

To do this, we added a submodule of type Global Parameters (GP). This type of submodule can only exist once per module and it is advisable to place it at the top to make it clear that its filters affect the rest of the submodules.

In the configuration of this submodule:

- We define the title of the section and the text of the button that will apply the filter.
- We create the parameters (Start Date and EndDate) and select the data type:
    - DateTime to close.
    - String for text.
    - Int/Double for numeric values.
    - CheckBox for Boolean options.
- We set the input control (in this case, DateTimePicker) and set default values, for example, last 30 days.
- We save the changes to activate the submodule.




**Benefits of Global Parameters**

- **Consistency:** all submodules display information under the same filtering criteria.
- **Efficiency:** you avoid setting the same parameters in each submodule.
- **Flexibility: Any** changes to the global parameter automatically impact the entire page.

## Conclusion

In this unit we have learned how to incorporate and organize submodules within the Data Services module and how to configure global parameters that unify the visualization of information. In the next unit, we will advance in the connection of these submodules with external data sources, applying global filters to enhance the customization and accuracy of our reports.

# Unit 2: Creating a SQL Query and Configuring Parameters

Every report starts with a solid foundation: the data we are going to analyze. In this unit we will learn how to create SQL queries, apply parameters, and organize the information in a way that suits our specific needs. Even if you haven't worked with SQL before, we'll look at each step in a clear and understandable way.

## What is a SQL Query?

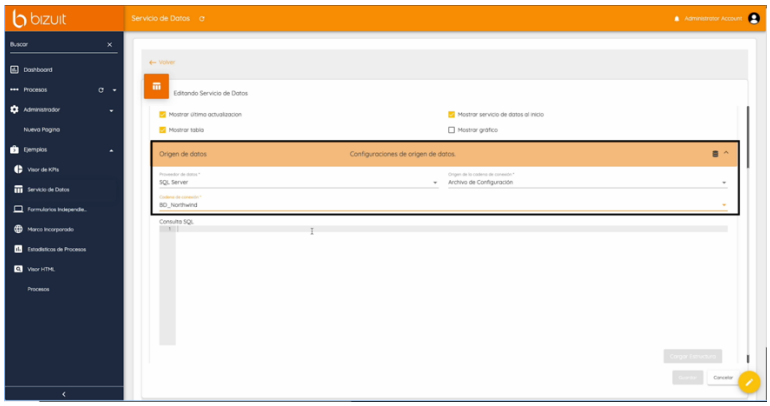
We can think of a SQL query as a recipe: we choose the ingredients (the data) and define how to combine them to get a useful result. For example, if we want to know who our best customers are, a SQL query can calculate the total purchases of each one and order them from highest to lowest.

**Step 1: Configure the Data Provider**

To begin, we select the submodule where we will display the information and define where we will obtain it from:

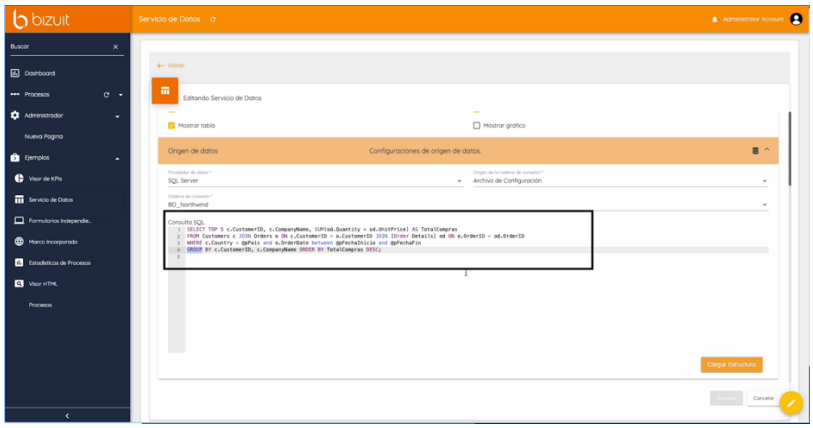
- **Data Provider**: SQL Server, OLEDB, ODBC, BigQuery or BIZUIT Processes.
- **Connection String Source**: Text (manual) or Configuration File (predefined).
-

In our example we will use SQL Server with Configuration File, selecting the string BD_Northwind, which points to our example database.




**Step 2: Write the Inquiry**

With the connection ready, we compose the query:

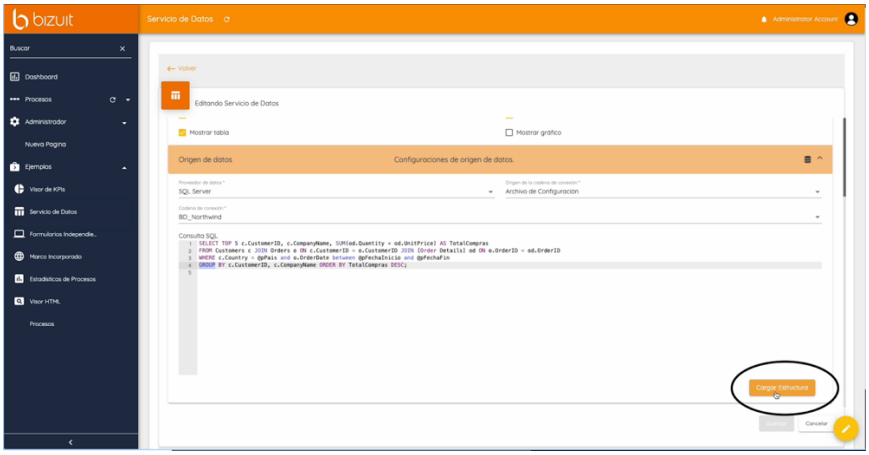



This instruction:

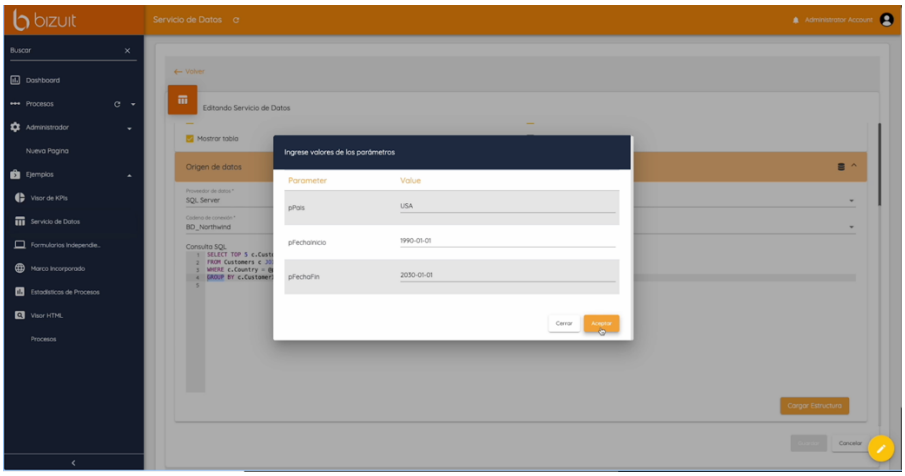
- **Select** key data (ID, customer name, and total purchases).
- **Relate** tables using JOIN.
- **Filter** by country and date range using parameters (@pPais, @pFechaInicio, @pFechaFin).
- **Group and sort** to show customers with the most purchases first.

**Step 3: Validate the Query**

By pressing Load Structure.



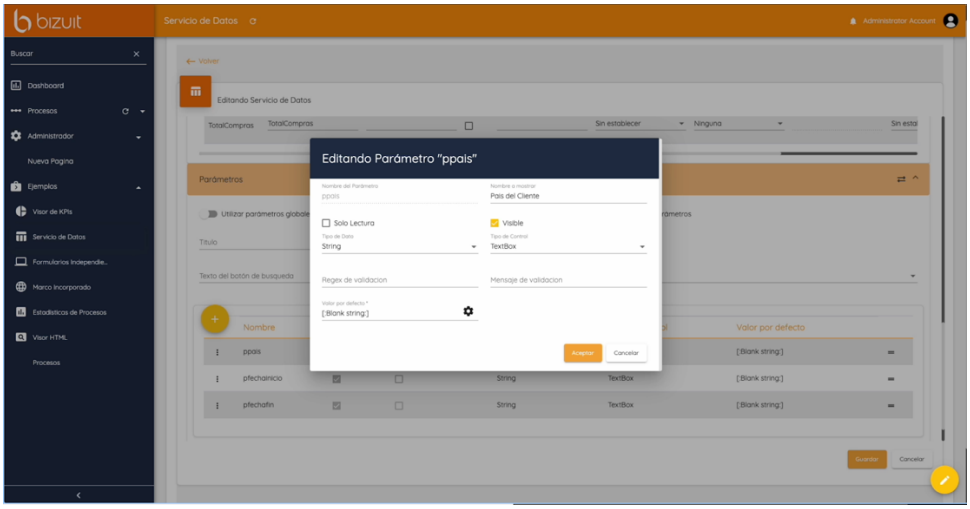

BIZUIT checks the syntax, requests values for the parameters, and displays the results. This step ensures that the query is ready to run smoothly.




**Step 4: Configure Parameters**

The parameters make the report dynamic. BIZUIT automatically detects them when it finds variables preceded by @ in the query. From the configuration section we can define:

- **Display name** (e.g. "Customer Country").
- **Visibility** (visible or hidden).
- **Editing** (editable or read-only).
- **Data Type**: Text, Number, Date, or Boolean.




**Step 5: User Controls**

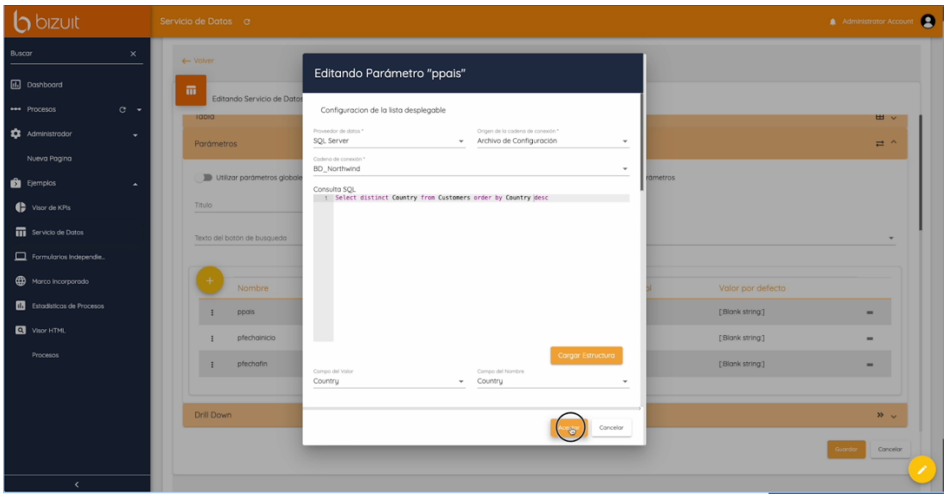
We define how the user will provide the parameter value:

- **TextBox**: free entry.
- **DateTimePicker**: Dates and ranges.
- **CheckBox**: Yes/No options.

- **ComboBox**: List of values obtained from another query.

### Step 6: Link to Global Parameters

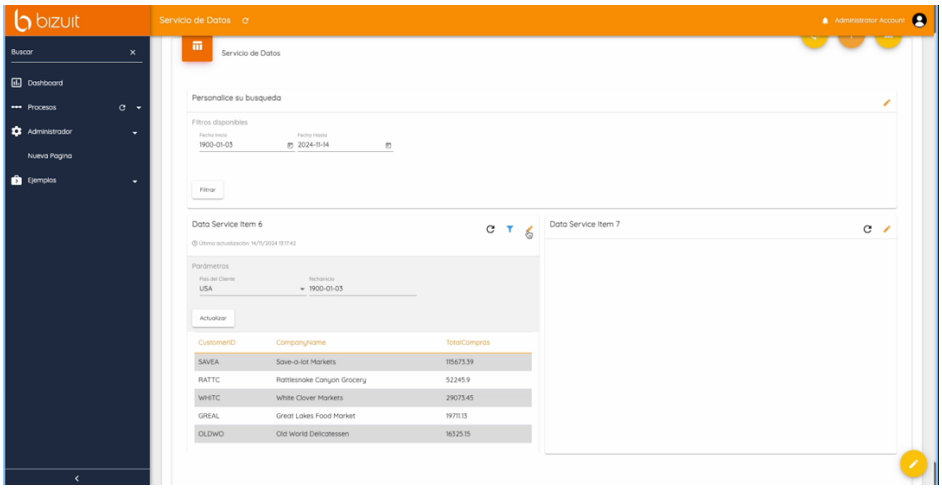
For the submodule to use the global parameters, the names must match. If we use @pFechaInicio in the query and the global one is called StartDate, we must unify them. Once corrected and validated, the global filters will be applied automatically.




### Final Validation

We tested the query with different scenarios:

- U.S. customers in the last 30 days.
- Customers since the beginning of the year.




We adjusted visibility and parameter editing permissions to simplify the user experience.

## Conclusion

In this unit we set up a submodule to get data using SQL queries and parameters in BIZUIT Dashboard, learning how to connect it to the data source, optimize its query, and link it with global filters.

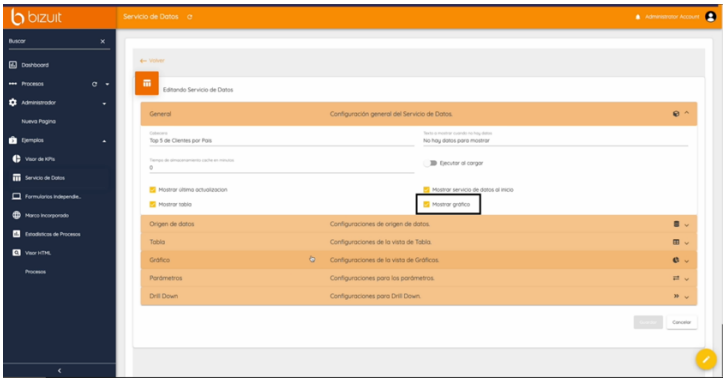
In the next unit we will transform this data into interactive graphs, taking the information from the technical to the visual.

# Unit 3: Setting Up Charts and Tables

A picture is worth a thousand words, and in data analysis, graphs are key allies to identify trends and patterns in a visual and intuitive way. In this unit we will learn how to create and customize charts within the BIZUIT Data Services module, configuring every detail to make them clear, useful, and attractive.

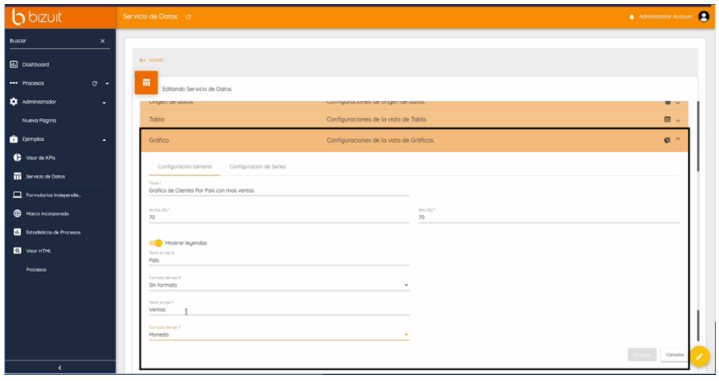
**Step 1: Basic Chart Settings**

Enter the editing of the submodule and activate the Show Graph option. This will enable the Chart section in the settings.



There:

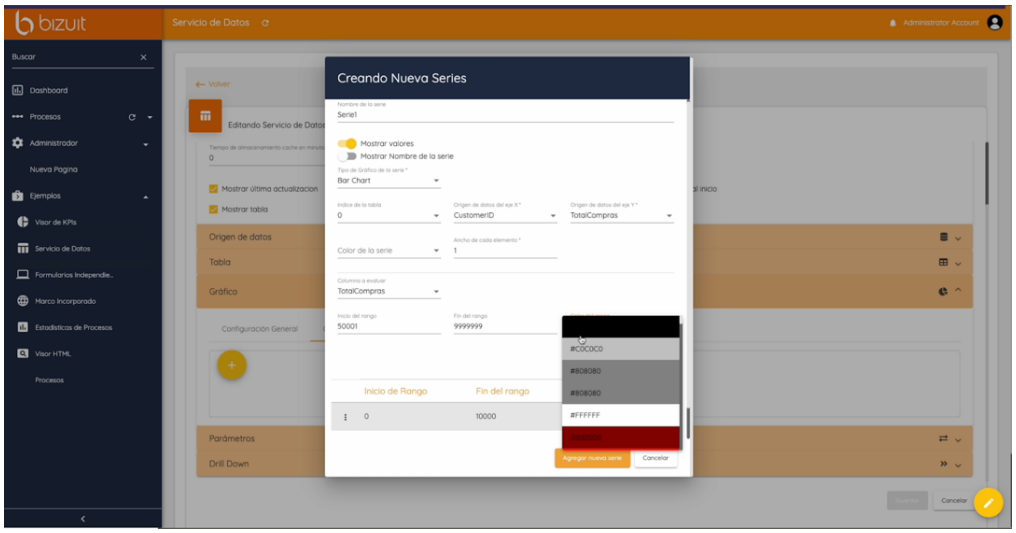
- We define a clear and relevant title, for example: *Customers by country with the most sales*.
- We adjusted size (width and height in %) to integrate it into the overall design.
- We set up X and Y axes: X will show the category (Customer) and Y the numerical value (Total Sales).



### Step 2: Set Up Series

Each chart can contain one or more **data series**, representing different sets of information.

- Select the **type of chart** (bars, lines, pie, etc.).
- We indicate which column of the query will feed the **X-axis** and which the **Y-axis**.
- We adjust bar or line thickness, main color, and, if necessary, colors by value ranges on the Y-axis to highlight critical data.




### Step 3: Validation and Testing

We test the chart and make adjustments to the size or format. We can decide to show only the graph, only the table, or both in combination.

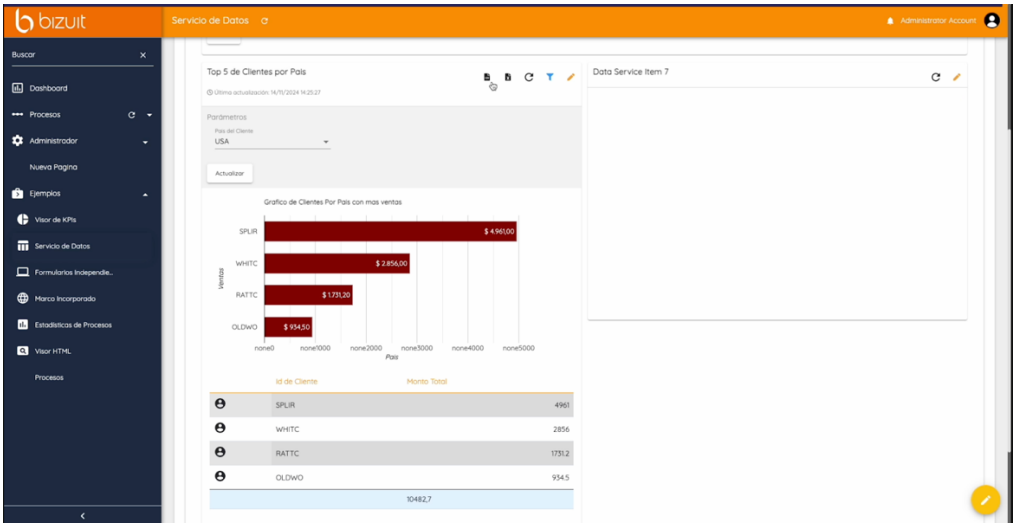
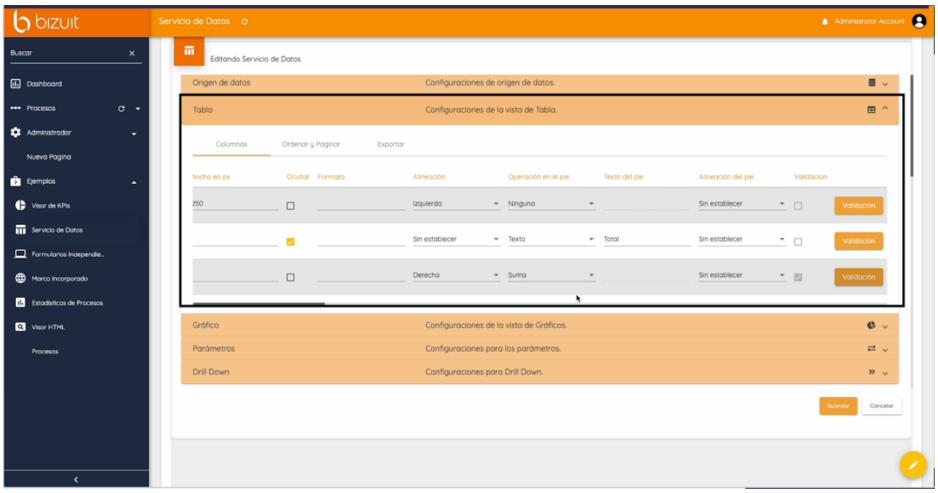


**Table Settings**

If we choose to also display the information in tabular format:

- In the **Table** section , we set up headers, column visibility, formatting, alignment, and aggregate operations (for example, sums).
- We define highlighting rules: change the background color or show an icon if a value meets a certain condition.
- We set up pagination, sorting, and export options.




## Conclusion

In this unit we learned how to transform data into clear and attractive visualizations, adjusting titles, axes, series, colors, and table formatting to highlight key information. With these tools, we can now generate visually striking and functional reports in BIZUIT Data Services, optimizing the interpretation and value of our data.

In the next unit we will continue to expand our skills so that we take our reporting to the next level.

# Unit 4: Configuring Drilldowns

In this unit we will learn how to use one of the most powerful functions of the BIZUIT Data Services module: drilldown. This tool allows us to connect submodules within the same page or between different pages, so that, with a single click, we can drill down into the information and explore it dynamically. The goal is to create interactive reports that make it easier to navigate and analyze data.

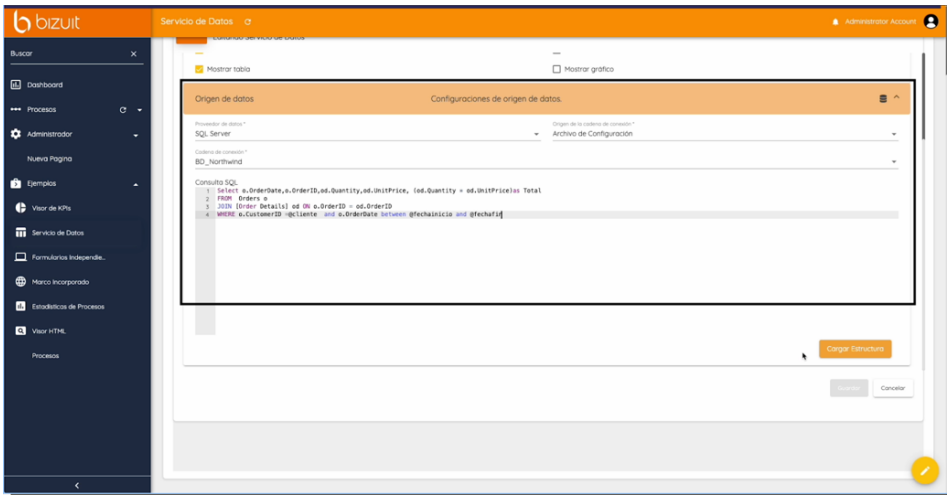
**Step 1: What is a Drilldown?**

A drilldown is like opening a second layer of information from a specific piece of data. For example: if we have a sales report by customer and country, when we click on a customer from a certain country, we will automatically see the breakdown of all their sales in that context, without leaving our workflow.

**Step 2: Prepare the Target Submodule**

The drilldown needs a submodule that receives and displays the detail of the information selected in the main submodule.

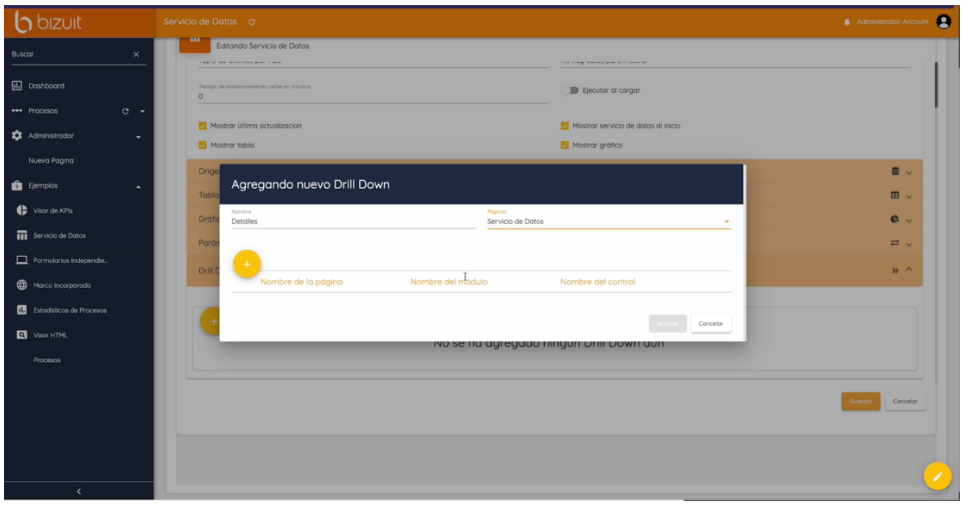
Example SQL query to display sales by customer and date range:




- We set up the data provider, connection string, and load the query.
- BIZUIT will check the syntax and ask for values for the parameters.
- We customize headers and save the settings before moving on to the main submodule.

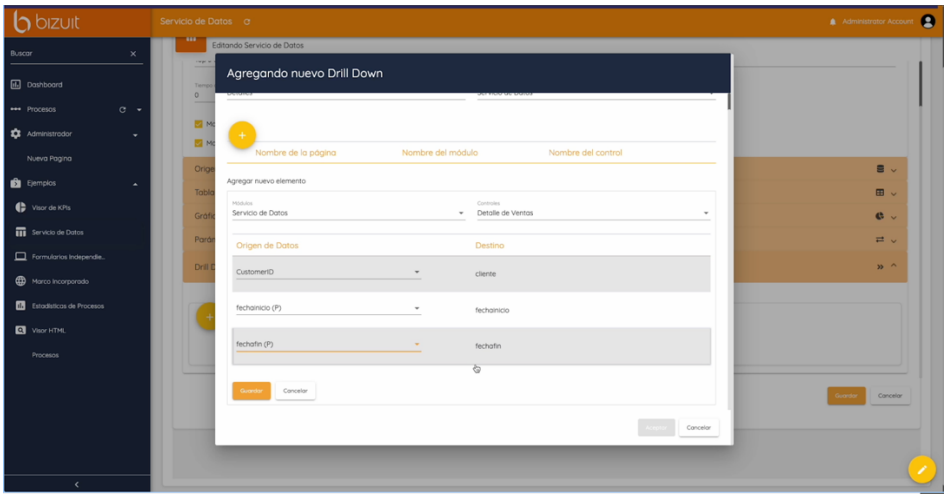
**Step 3: Set up the Drilldown**

1. In the main submodule, we open the DrillDown section and add a new one.
2. We assign a name for the button that will activate the detail.

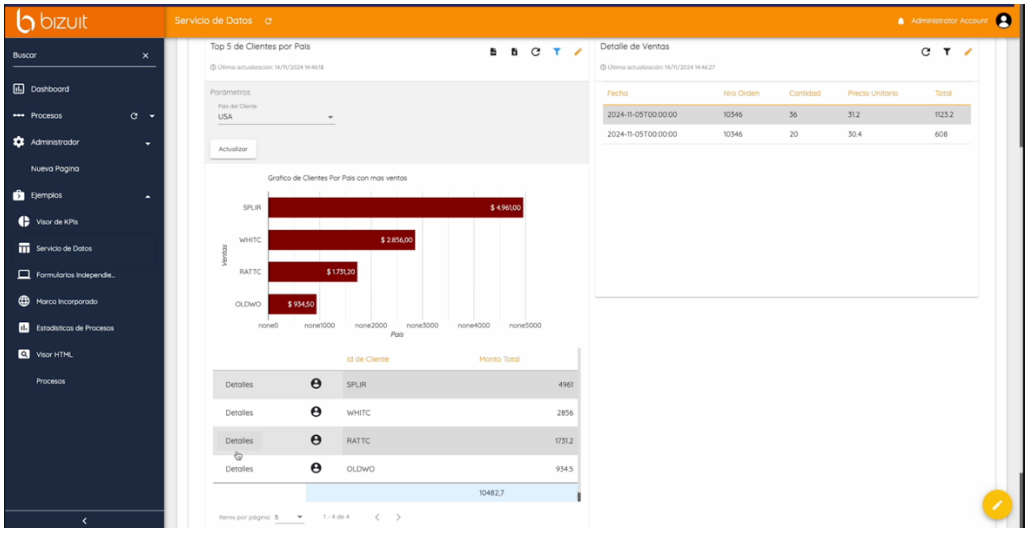



3. We select the page and the Data Services module that contain the target submodule.
4. We indicate the target submodule.
5. We map parameters:

    - Customer will take the value from the *CustomerID* column of the selected row.
    - Start Date and End Date will be obtained from the parameters applied in the source submodule.




We save and test that, when selecting a customer in the main report, the corresponding detail is displayed in the target submodule.




## Conclusion

We now know how to set up a drilldown in BIZUIT, connecting submodules to deliver interactive and easy-to-explore reports. With this technique we can build more fluid, detailed analysis experiences adapted to the needs of your users.

# Chapter Summary

**BIZUIT**'s Data Services module is presented as a comprehensive solution for consulting, processing and visualizing information, transforming data into clear, dynamic and interactive reports that facilitate analysis and decision-making.

Throughout this chapter, we go through, step by step, the key functionalities offered by this tool. In **Unit 1**, we looked at how to incorporate the module into the Dashboard, create and organize submodules, and configure global parameters to unify filters across the page. This approach allowed us to achieve consistency, efficiency and flexibility in the presentation of information. In **Unit 2**, we learned how to configure the data provider and source, write optimized SQL queries, and use parameters to filter and customize reports. We also looked at how to link these parameters to global ones and offer input controls—such as drop-down lists, date pickers, or text fields—to facilitate user interaction. In **Unit 3**, we focused on visual representation, transforming the data into customizable charts—with titles, axes, series, and colors—and configurable tables—with headings, formats, highlights, aggregated operations, and export options—which allowed us to identify patterns and improve visual interpretation. Finally, in **Unit 4**, we implemented **drilldown** functionality to connect submodules, drilling down into the information with a single click. We set up target submodules, map parameters, and create a smoother, more contextual analysis experience.

Together, all these capabilities give us the possibility of creating robust, aesthetically careful and highly interactive reports, capable of adapting to the most demanding analysis needs. With this foundation, we are ready to take the visualization and exploitation of data in BIZUIT to an advanced level.