Chapter 11: Customizing the BIZUIT User Interface

Welcome to this chapter on Dashboard customization and optimization in BIZUIT. We will learn how to transform the Dashboard into a visually attractive, functional tool aligned with our company's identity.

We will explore everything from the configuration of colors and themes to the creation of custom modules and the use of independent forms, discovering all the aspects that allow us to adapt the BIZUIT Dashboard to our specific business needs.

We'll start with visual customization, learning how to select colors and themes that reflect the company's brand, and setting up logos and icons that reinforce that identity. By adapting these elements, we will achieve a work environment that feels own and familiar to our users, improving their experience and creating visual coherence.

Next, we will focus on the available modules, especially the Built-in Framework, the HTML Viewer and the Processes module, which allow us to integrate external and custom content, as well as group processes according to their functionality. With the Built-in Framework module, we can view reports, documentation, and more without leaving BIZUIT. The HTML Viewer module offers us the flexibility to add metrics, shortcuts and any content adapted to our needs, turning the Dashboard into a true centralized and efficient information center. In addition, the Processes module allows our users to quickly access the most used processes from the Dashboard itself.

Standalone forms are another key component in BIZUIT. In this unit, we'll look at how we can use them not only to capture data, but also to visualize it in structured formats, such as grids and complex screens. These forms can be shared via URLs, with or without authentication, or even integrated into other sites, making them a powerful solution for data management and creating custom interfaces.

Subsequently, we will explore the plugin functionality, which allows us to create fully customized modules programmatically. These modules can perform complex functions, integrate with external systems, or guide the user through specific steps, such as a validation wizard. In this unit we will learn how to leverage this tool to develop unique features that optimize our company's specific processes.

Finally, we will address optimization strategies. We'll look at how to maintain a balance between aesthetics and performance, sharing best practices for organizing visuals, improving navigation, and making the Dashboard intuitive and easy to use.

These tweaks are essential to improving the user experience, ensuring that every interaction is fast and efficient. In this chapter, we will go step by step by discovering how to customize and optimize the BIZUIT Dashboard, adapting it to the needs of each team and creating a visually coherent and functional work environment.

We hope that this series will become a useful tool to transform our Dashboard into a powerful platform oriented to the objectives of our company. Start!

Ideal Audience

This course is aimed at professionals who participate in digital transformation projects and have an interest in the design and customization of user interfaces within BIZUIT. It is recommended to have basic knowledge in: Configuration of UX/UI information systems Experience in designing interfaces for business applications.

Objectives

1. Set up custom colors, themes, logos, and icons in BIZUIT: Learn how to tailor the look and feel of the BIZUIT Dashboard to reflect your organization's visual identity.

2. Use and configure the modules available in BIZUIT: Discover how to incorporate and manage modules in the Dashboard for different types of applications.

3. Extend Dashboard functionality with custom modules: Learn how to create custom modules using components to meet specific business needs.

4. Introduce the use of standalone forms: Explore how to develop flexible and custom interfaces using standalone forms in BIZUIT.

5. Customize the user experience in BIZUIT: Optimize the visual and functional interaction of the environment, adapting the platform to the needs of your organization.

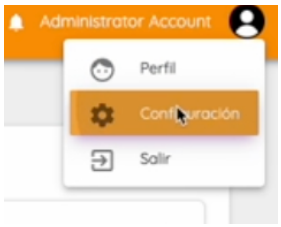
Unit 1: Setting Up Colors, Themes, Logos, Icons, and Viewing Process Forms

In this unit, we'll explore how to customize the BIZUIT Dashboard to reflect our company's visual identity. Personalization not only improves the user experience, but also strengthens the corporate image within the platform.

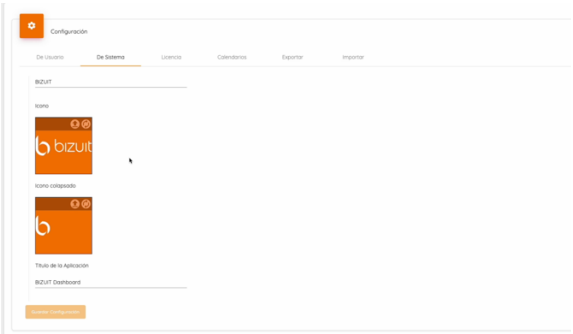
We'll set up colors, themes, icons, and other visual elements, making sure the work environment is engaging, functional, and aligned with our brand.

Customizing the Dashboard in BIZUIT: Configuring Colors and Themes

Step 1: Access the Settings Panel
To start, we access the configuration module from our user. Select the "Settings" option and go to the "System" tab, where you will find all the options to customize the appearance of the Dashboard.



It is essential to have administrator permissions or the necessary access, as these changes will affect all users of the system.



Step 2: Customizing Icons

At the top of the settings, we find the options to modify the main icons:

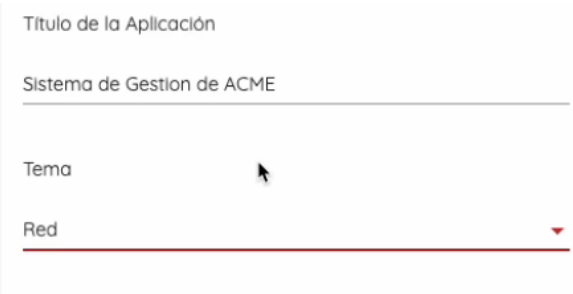
- **Main Icon**: This is the logo visible in the sidebar when the menu is expanded. To customize it, we simply upload the desired image.
- **Collapsed Icon**: This logo is displayed when the menu is minimized. A simplified version is recommended for clarity.

To ensure a clean and professional design, it is ideal to use PNG images with a transparent background and good resolution.

Step 3: Setting Up the Dashboard Title and Theme

A little further down, we find the option to define the Application Title, which will appear in the browser header and in other key sections. We may use our company name or a clear description, such as *"Corporate Control Panel"*.

Then, we select the Dashboard Theme, which defines the main colors of the system. If we have specific corporate colors, this is the opportunity to apply them to maintain the visual coherence of our brand.

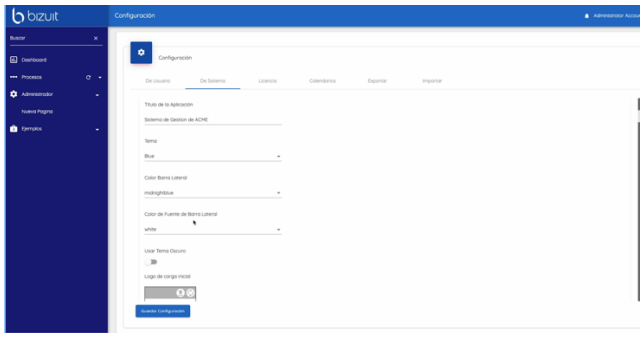


Step 4: Sidebar Customization

The sidebar is a highly visible element and we can adjust it according to our corporate colors:

- **Sidebar Color**: We define the background of the bar where the menus are located.
- **Sidebar Font Color**: We set the color of the text within this bar. It is crucial to ensure good contrast to improve readability.

For example, if we use a dark background, such as navy blue, the ideal would be white text or in light tones so that it stands out correctly.

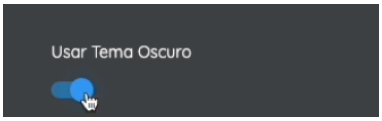


Step 5: Activating the Dark Theme

If we prefer a more modern design or need a more comfortable interface for low-light environments, we activate the Dark Theme.

This option changes the colors of the system to shades that are more soothing to the eye, which is useful if users work for long hours on the platform.

Each user can overwrite these settings according to their personal preferences.



Step 6: Customizing the Cargo Logo

The Initial Load Logo appears while the system is booting. Here we can upload our company logo or a custom design to reinforce the brand identity from the first moment.

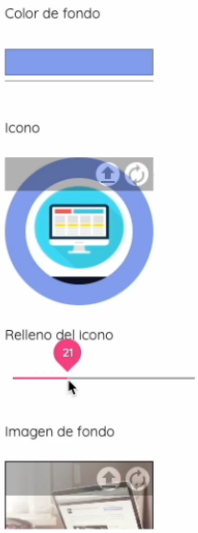
We can also configure the favicon, which is the small icon visible in the browser bar. Making sure that these elements are visually appealing will contribute to a better presentation of the system.

Step 7: Setting the Background and Style of the Login Screen

The login screen is another opportunity to reflect our visual identity. We can configure:

- **Background Color**: We choose a solid color according to our brand.
- **Login icon**: We load a logo that will appear on the home screen. We can adjust its size to ensure proper positioning.
- **Background Image**: If we prefer a more personalized design, we upload a background image. It is recommended that she be discreet and professional to avoid distractions.

- **Background Style**: We define whether the image is displayed in its original size, adjusted or covering the entire screen.



Step 8: Process Customization

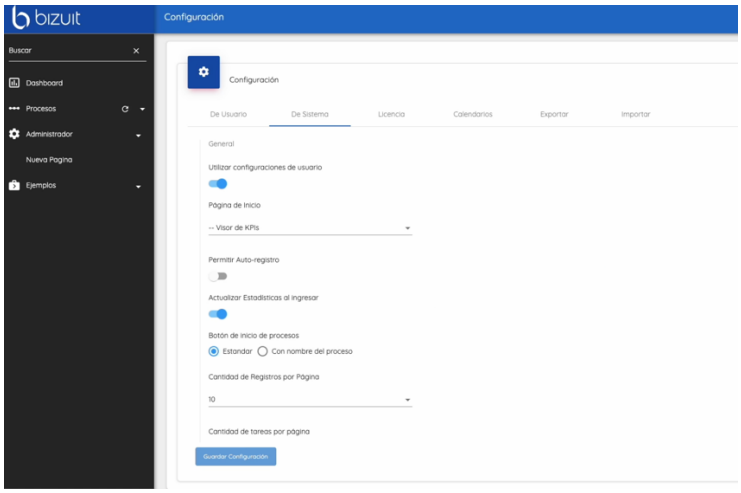
In the side menu of the Dashboard, the processes are presented with an icon and a color configurable from BIZUIT Designer at the time of publication.

We can choose default or custom icons, ensuring that they contrast with the overall design and make it easy to identify each process. Once published, these changes are automatically reflected in the Dashboard, improving organization and navigation.

Step 9: Defining the Homepage

We can set the default home page of the system. By default, this will be the Dashboard, where process statistics are displayed.

If we want the statistics to be updated at each entry, we activate the option "Update Statistics when entering". It is important that the selected page is accessible to all users, otherwise they could receive an authorization error when entering.



Step 10: Final Testing and Adjustments

Once all the changes have been applied, we perform a visual review of the Dashboard:

- We browse through the sections to check the consistency of colors and visual elements.
- We adjust contrasts and details, ensuring correct readability.
- We verify that buttons and links are identifiable and functional**.**

Small modifications can have a big impact on the user experience, so it's a good idea to test before finalizing the setup.

With these customizations, our BIZUIT Dashboard will not only be a functional tool, but also a reflection of our company's visual identity. A well-fitting design improves the user experience, strengthens the connection to the platform, and projects a professional image.

Process Form Display Customization

Now we'll explore the customization options in BIZUIT to configure the way process forms are opened and displayed. Adjusting these settings allows us to optimize the user experience and tailor it to the needs of our organization.

We'll discuss the different view styles, dashboard settings, and other advanced options that will help us make using forms more efficient and dynamic.

Step 1: Configuring the Forms View

Within the configuration section we find three main options to define how the forms of the processes will open by default:

- **Dialog**: This is the default option. Open the form in a focused pop-up, ideal for quick and focused tasks.
- **Window**: Allows you to open the form in a full window, providing more workspace, especially useful for long or multi-section forms.
- **Split Panel**: Displays the form in one part of the screen while other elements, such as the task tray, are visible. This option is ideal for those who need to interact with multiple items at once.

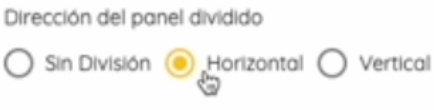
Selecting the right option will depend on the type of process and the level of interaction required by each user.

Step 2: Split Panel Direction

If we opt for the Split Panel view, we can further customize its orientation:

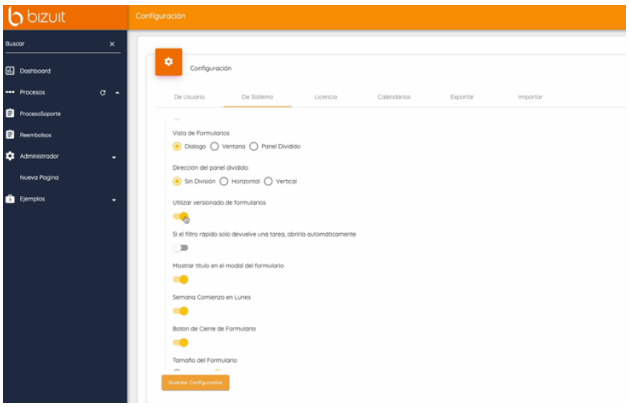
- **No Division:** Allows you to work with the full view of the form.
- **Horizontal:** Divide the screen into two sections, placing the form above and the other elements below.
- **Vertical:** Separates the screen into two columns, with the form on one side and the other elements on the other.

Each user can modify these settings individually for each activity within each process, ensuring that the visualization fits their specific workflow.

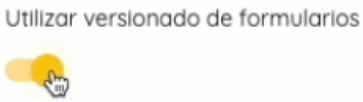


Step 3: Enabling Advanced Features

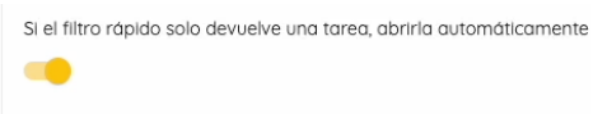
In addition to view and orientation settings, BIZUIT offers advanced options to further customize the user experience:



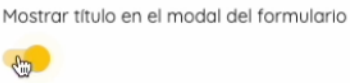
- **Form Versioning**: It allows us to manage multiple versions of a form, ensuring that each user works with the appropriate version according to the version of the process.



- **Automatic Form Open**: If the Quick Search Filter returns a single task, this option automatically opens the corresponding form, saving time on navigation.



- **Show Title in Form Modal**: We can turn on or off the display of the title within the pop-up window, depending on whether we want a cleaner interface or if we prefer to display the name of the form for clarity.

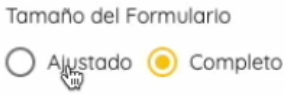


These advanced features can make all the difference in terms of efficiency and ease of use.

Step 4: Setting Form Size and Behavior

To ensure proper presentation, we can adjust the size and behavior of the form:

- **Form Size**:



- *Adjusted*: Adapts the size of the form to the content, avoiding unnecessary spaces.
- *Complete*: Expands the form to take up all available screen space, ideal for longer forms.

- **Form Close Button**: This option ensures that users always have a visible button to close the form clearly and quickly, improving navigation and usability.

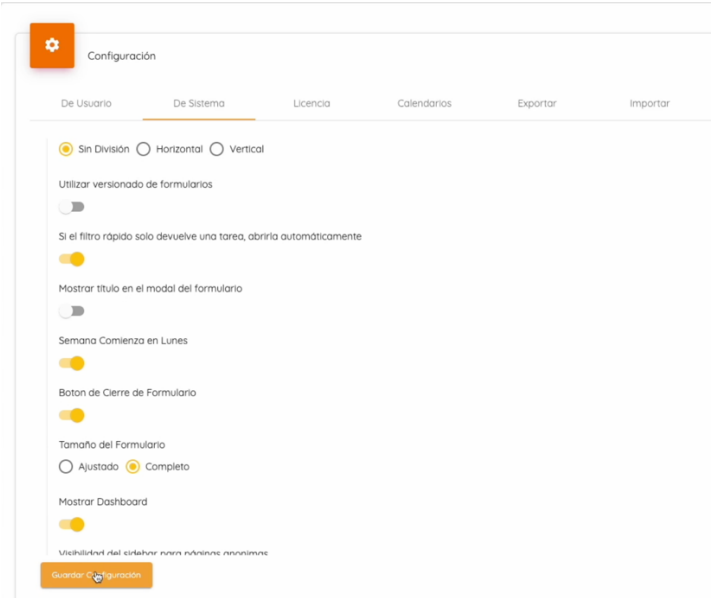
These settings allow us to define how users will interact with forms within the system.

Step 5: Final Testing and Adjustments

Once all the options are configured, it is essential to perform tests to ensure that the user experience is optimal. To do this, we verify that:

- The configured views are functional and adapt to the needs of each process.
- Split panels are correctly oriented and do not cause inconveniences in the visualization.
- Advanced options, such as auto-open or title visibility, work as expected.

If we detect any necessary adjustments, we make the corresponding modifications to ensure the best possible experience.



Thanks to these customization options, BIZUIT allows us to adapt the user experience in the forms according to the needs of our organization.

By leveraging these tools, we improve team efficiency and offer a clearer, more professional interface.

Conclusion

In this unit, we learned how to customize the BIZUIT Dashboard to reflect our company's visual identity. We set up colors, themes, icons, and logos, making sure the interface is consistent with our brand. We tweaked key elements such as the sidebar, dark theme, loading logo, and login screen, optimizing the appearance and usability of the system.

We also explore process setup and the homepage, allowing users to efficiently access relevant information. Finally, we carry out tests to verify that the changes applied are functional and aesthetically adequate.

In addition, we analyze how to customize the display of forms in BIZUIT, choosing between different opening options: pop-up dialog, full window, or split panel.

We also set up dashboard targeting and enabled advanced features like form versioning and auto-opening for specific searches. We adjust the size and behavior of forms to improve navigation and ensure an optimized user experience.
Thanks to these tools, we make the work environment more efficient, intuitive and aligned with the needs of our organization.

Unit 2: Using the Modules Available in the BIZUIT Dashboard

Welcome to this unit, where we will explore the standard modules that BIZUIT offers in its Dashboard. In particular, we'll focus on three key tools: Built-in Framework, HTML Viewer, and the "Processes" module.

These modules allow us to integrate external content, customize the interface with rich information, and provide direct access to essential tasks and processes. With these options, we can adapt the work environment to the needs of our company and optimize the user experience.

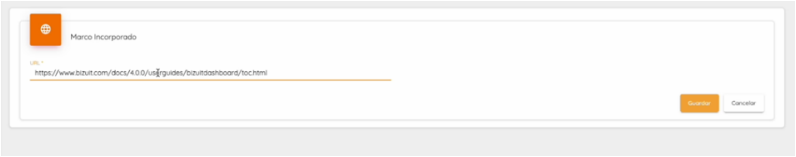
Step 1: The Built-in Framework Module
The Built-in Framework module allows us to display external content within the BIZUIT Dashboard, such as web pages, reports, or third-party tools. This is useful when we want users to access external information without leaving the platform, ensuring that they are always working with up-to-date data.

Examples of use:

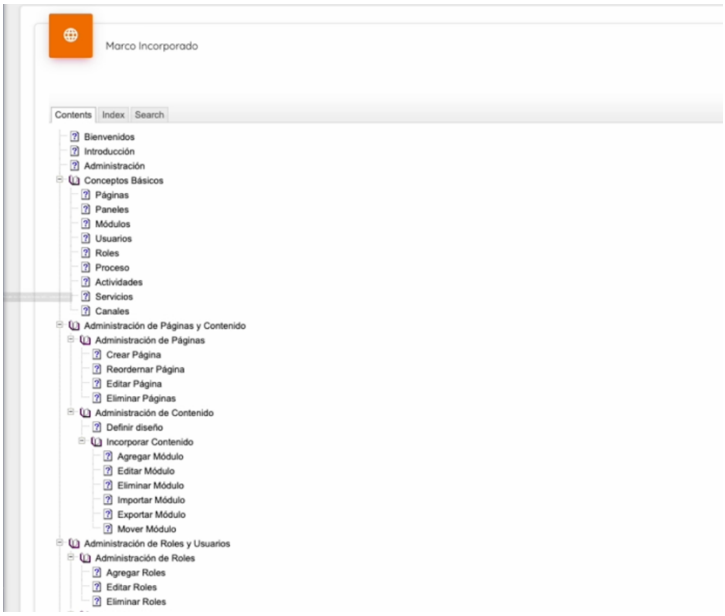
- **Data analysis dashboards**: We can integrate tools such as Power BI or Google Data **Studio** so that reports are accessible directly from the Dashboard.
- **Documentation of procedures**: If we store manuals on external platforms, we can display these guides in the Dashboard for quick access.

How to set it up:

1. We added the **"**Built-in Framework" module in a section of the Dashboard.
2. We edit the module and add the external URL we want to load.



3. We save the changes and verify that the content is displayed correctly.



Step 2: The HTML Viewer Module – Customization and Rich Content

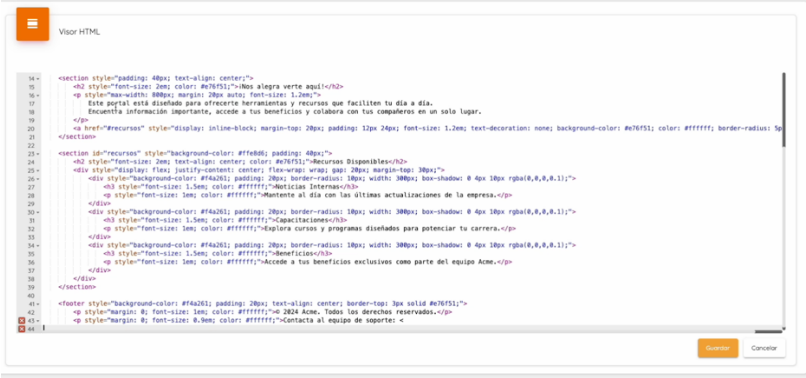
The HTML Viewer module allows us to add custom visual content within the Dashboard, using HTML elements. With this tool, we can improve the user experience by adding dynamic content tailored to our company's needs.

Examples of use:

- **Internal News Dashboard**: We publish announcements, announcements, and reminders directly in the Dashboard.
- **Calendar of events**: We show training, meetings or holidays with different colors depending on the type of event.
- **Branch map**: We integrate an interactive map with the location of offices and links to contact information.

How to set it up:

1. We added the "HTML Viewer" module on the Dashboard page.
2. We edit the module and insert the HTML content we want to display.



3. We save the changes to view the updated content.

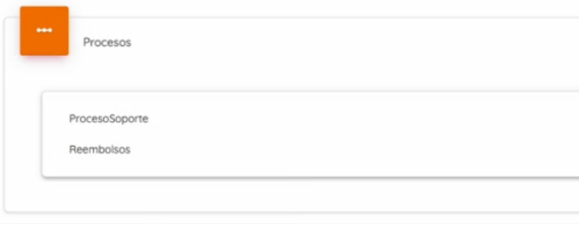
Step 3: The Processes module – Direct access to specific tasks

The "Processes" module facilitates access to the most frequently used tasks, allowing users to start processes directly from the Dashboard without having to search in the side menu.
This improves efficiency and reduces the time needed to execute key tasks.

How to set it up:

1. We added the "Processes" module in the Dashboard.

2. We select the processes we want to include, for example, "Support" and "Refunds".
3. We save the changes so that the selected processes are available.



Advantages of the Processes module:

1. Improve accessibility by concentrating the most frequently used processes in one place.
2. It reduces the clicks needed to start a process, optimizing the user's time.
3. It allows flexible customization according to the needs of the team or work area.

Conclusion

In this unit, we explore how the Built-in Framework, HTML Viewer, and Processes modules can be used to customize and optimize the BIZUIT Dashboard. Not only do these tools improve the user experience, but they also make it easier to access information and streamline workflows.

With the knowledge gained in this unit, you can now customize the BIZUIT Dashboard to become an operations center completely tailored to your organization's needs.

In the next unit, we'll explore advanced capabilities such as creating more complex interfaces to solve challenges specific to your business.

Unit 3: Introduction to Standalone Forms in BIZUIT

In this unit, we will explore standalone forms in BIZUIT, a key tool for those looking for a flexible and efficient solution to collect and manage information in a personalized way.

Unlike other elements integrated into workflows, these forms work autonomously, allowing us to capture and display information without being tied to a rigid process. In addition, their ability to connect with customizable data sources and execute advanced tasks makes them a versatile solution for multiple scenarios.

Standalone forms in BIZUIT are customizable interfaces designed to collect and visualize information without the need to be integrated into a main process flow. Not only do they allow us to capture data, but also to display information in grids and build complex screens according to our needs.

In addition, these forms can be connected to various data sources, allowing us to:

- Invoke specific processes within BIZUIT.
- Execute calls to REST APIs to integrate external services.
- Perform SQL statements on databases, extracting or manipulating information directly.

Thanks to this combination of functionalities, independent forms become a powerful tool for centralizing data, executing complex actions and designing interfaces adapted to different teams and projects.

Key Benefits of Using Standalone Forms

These forms are completely customizable, allowing us to add or remove fields as needed. In addition, they can be configured to display data in grids, which is useful for generating interactive reports or visualizing structured information.

They also offer the option of interacting with databases using SQL statements, allowing information to be managed in real time.

URL Exposure and Authentication Options

One of the great benefits of standalone forms is that they can be accessible through a URL. This means that we can invoke them from external links, third-party applications, or within other BIZUIT pages.

In addition, we can configure authentication levels according to the security requirements of each case.

Thanks to their integration with REST APIs, these forms can interact with external systems in real time, from querying an inventory to updating records in a database.

Independence from the Main Process Flow

Not relying on a predefined workflow, standalone forms can be used at any time and in any context.

This allows us to invoke specific BIZUIT processes directly from the form, without the need to go through the main structure of the platform.

They are ideal for applications such as registration forms, internal surveys or autonomous management tools.

Creating Complex and Custom Screens

In addition to its data collection and visualization function, standalone forms allow us to build complex interfaces with multiple sections, data grids, and advanced controls. For example, we can:

- Display information extracted from databases using SQL statements.
- Include interactive elements to update data in real time.
- Generate dynamic reports directly within the form.

Use Cases for Standalone Forms in BIZUIT

- **Business Visitor Registration:** We may use a separate form to record visitor data, the reason for their visit, and the person they will be meeting. This form can be integrated with REST APIs to automatically notify the host and generate access passes.
- **Support Incident Management:** Support teams can use separate forms for users to report issues or incidents. Thanks to its integration with APIs and internal processes, we can automate notifications to those responsible for the area, escalate problems and generate follow-up reports.
- **Tables and Dynamic Reports:** We can use separate forms to create reports in grids that are updated in real time using SQL queries. This is ideal for inventory dashboards, sales reports, or case tracking. In addition, we can configure buttons or links that invoke processes directly from the grid, such as updating records or generating additional reports.
- **Invoking Different Processes on the Same Screen:** Standalone forms allow us to integrate multiple interactive controls that run different processes within BIZUIT from a single screen. This functionality centralizes essential operations, improving productivity and avoiding the need to navigate multiple screens. For example:

    - A project management form with options to create tasks, assign resources, and close projects.
    - A sales dashboard with buttons to register customers, update orders, and generate invoices.

Now that we understand the concept and usefulness of standalone forms in BIZUIT, we'll give you a hands-on demonstration of how to set them up in the next section.

We'll explore creating custom fields, implementing validations, integrating with APIs, using SQL statements, and invoking multiple processes on a single screen.

Creating and Configuring a Standalone Form

Now, we'll conduct a hands-on demo where we'll set up a standalone form in BIZUIT from scratch. We'll learn how to create custom fields, apply validations, connect SQL data sources, and invoke processes within the platform.

We will also configure security and anonymous access options and see how to integrate the form into the Dashboard using the Independent Forms Viewer module with support for versioning.

In this practice, we will design a form to manage fruit order requests, incorporating advanced functionalities that will allow us to interact with BIZUIT's internal databases and processes. In addition, we will explore how to adjust access and security options, ensuring proper integration into the Dashboard and ensuring that updates to the form are automatically reflected thanks to versioning.

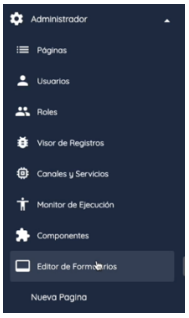
The form we will create will allow you to:

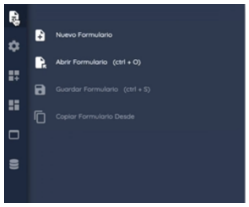
- Querying a SQL database to verify a customer's data.
- Invoke a BIZUIT process to record and assign the request to the appropriate team.
- Configure anonymous access for use without authentication.
- Generate a URL to make it easy to invoke from any environment.
- Integrate it into the Dashboard using the Independent Forms Viewer module.

This approach will allow us to cover a complete use case and demonstrate the advanced capabilities of standalone forms in a real-world environment.

Step 1: Create Custom Fields

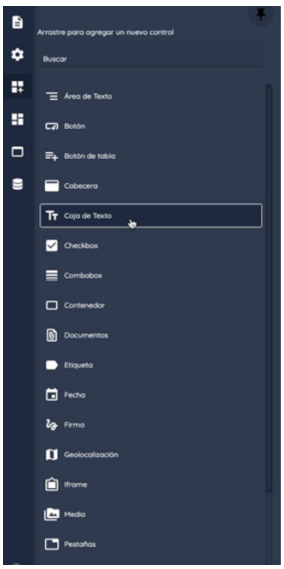
We'll start by going into the BIZUIT Forms Designer form editor and create a new form.



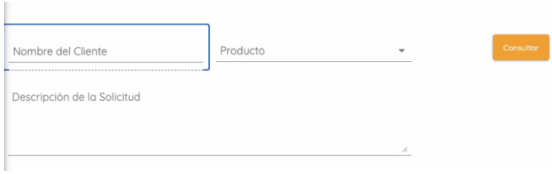


Next, we'll set up the necessary custom fields by dragging the following controls into the form design area:

- **Text field:** for the *Customer's Name*.
- **Drop-down list:** to select the *Product*, with a list of 5 fruits available.
- **Text Area Field:** for the *Request Description*.



In addition, we will add an action button that allows you to query a SQL database to check if the client exists before proceeding with the registration.

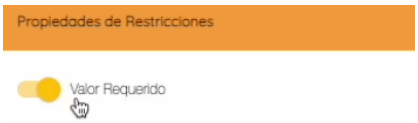


Step 2: Validations Configuration

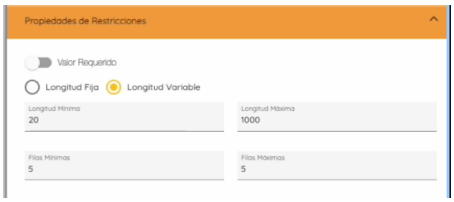
To ensure the accuracy of the data entered, we will configure the following validations in the property sheet of each control, which will be displayed when selecting the control in the design area

- The Customer *Name field* will be mandatory.

- The Product drop-down list may not be empty.

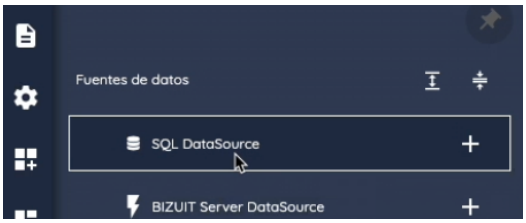


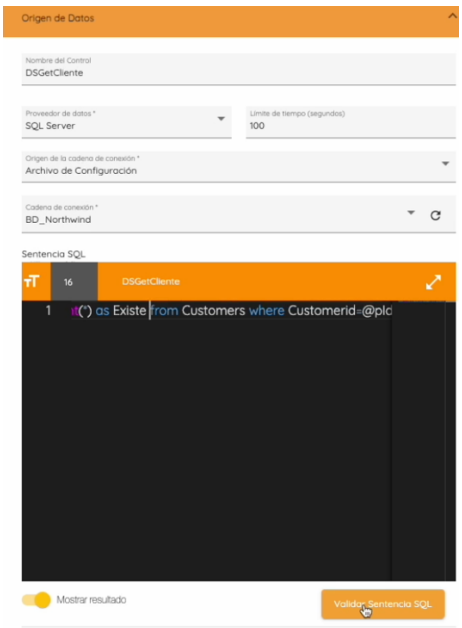
- The *Application Description* must contain at least 20 characters.



Step 3: Connecting to the SQL Data Source

We will configure a SQL query that will allow us to validate the existence of the client in the database, ensuring the consistency of the data captured.



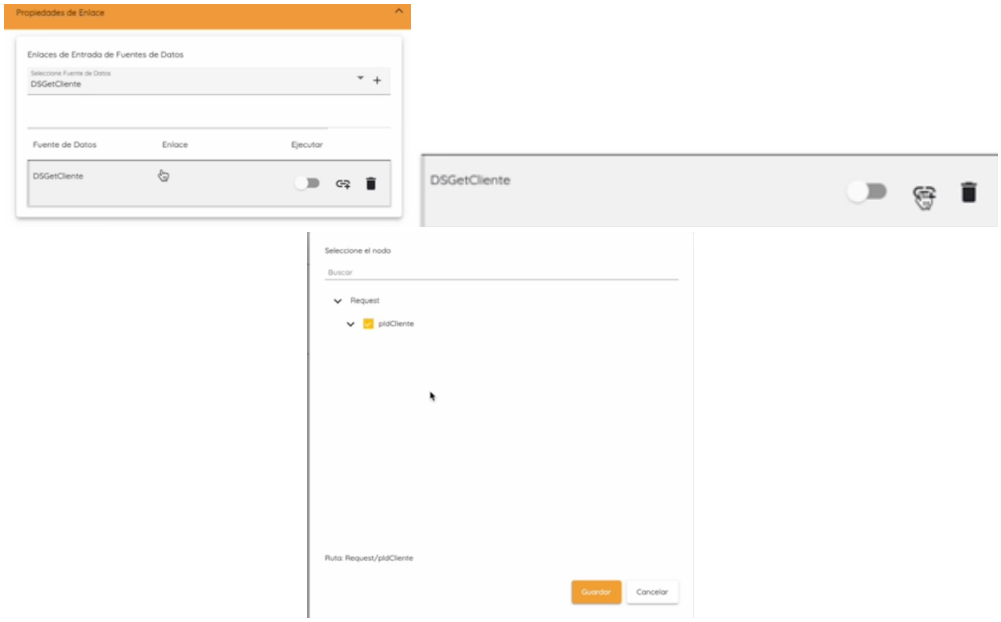


When the user enters the *Client Name* and presses the query button, a SQL statement will be executed that will search for the client.

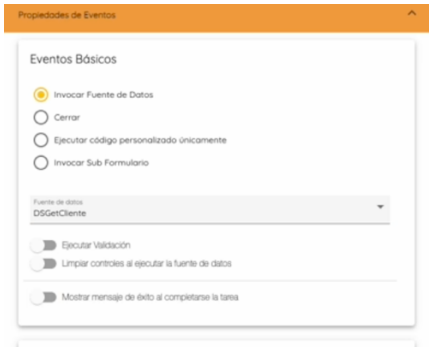
- If the client is not in the database, the form will display an error message.
- If the client exists, it will allow the registration to continue.

To do this:

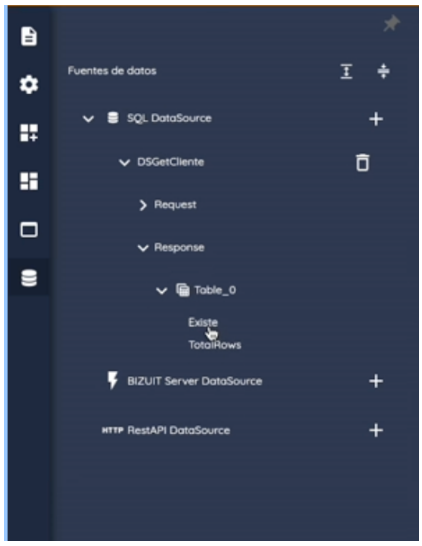
1. We associate the *Customer Name* field with the query input parameter in our secondary data source.



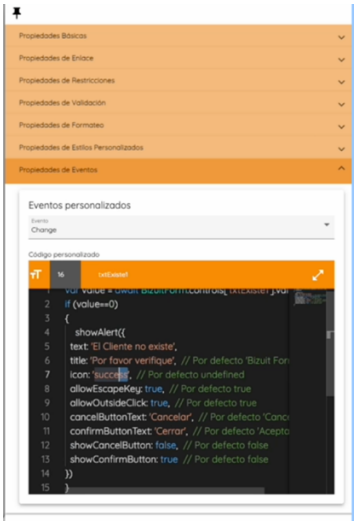
2. We configure the button to run that secondary data source.



3. We drag the "Exists" node of the response from the secondary data source into the form, which will allow us to display a text control with 1 (client found) or 0 (client not found).



4. We implemented a validation using custom code and tested its operation with the **"**Consult**" button**.



5. We hide the "Exists" control to improve the user experience.

Step 4: Invoking a BIZUIT Process

The form will invoke a process within BIZUIT to record and assign the captured request.

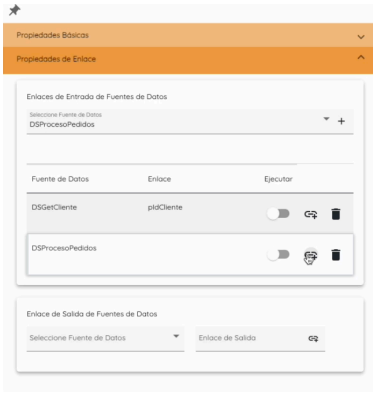
When the user presses the "Submit" button, the form will send the data to the configured process, registering the request in the system and automatically notifying the assigned team. To achieve this:

1. We create a secondary data source of type BIZUIT, pointing to the desired process.

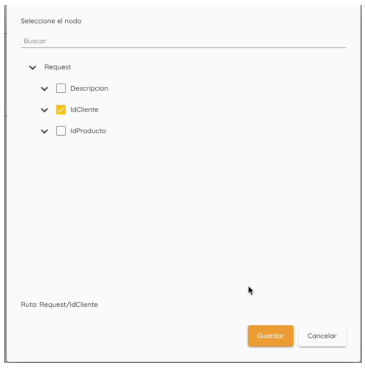
2. We associate the controls of the form with the parameters of the process:

   - We individually select each of the controls on the form.
   - In its binding properties we indicate the BIZUIT data source whose parameters we want to complete with the value of each control of the process as indicated in the following example for the "Customer Name" control:

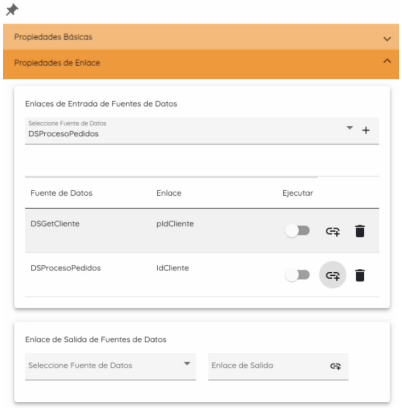
   - We add the data source to the list of Available Data Sources



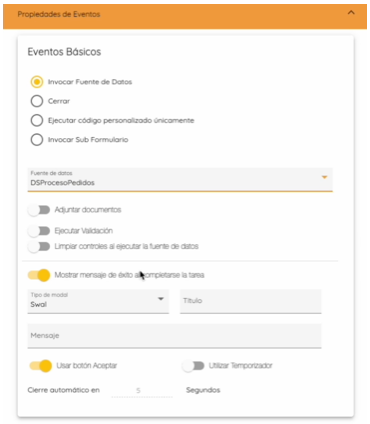
- We select the input parameter of the process that will execute the data source to which we want to send the value of the control we are configuring:



- Click on the "Save" button and repeat the steps for the Product and Description controls:



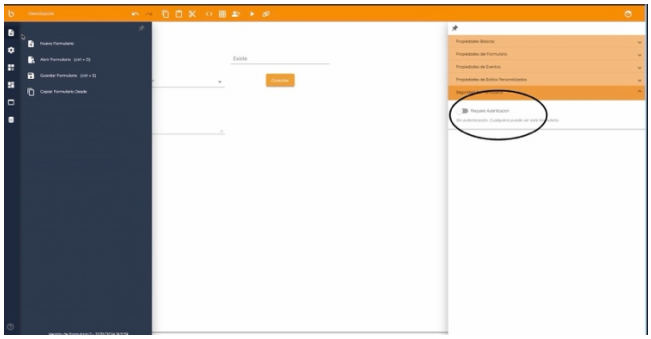
3. We set the **"**Submit**"** button to run the secondary BIZUIT data source.



Step 5: Security Settings and Access URLs

To make it easier to access the form, we will configure advanced security options.

- **We will enable anonymous access**, eliminating the need for authentication. This is useful when the form needs to be available to external or publicly accessible users.



- **We'll generate a unique URL** that will allow you to invoke the form from any environment, either by embedding it on an external site or by sharing the URL directly with users.

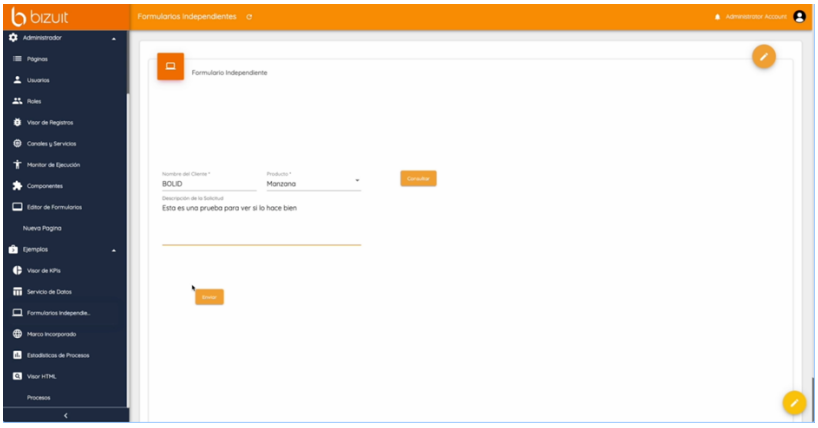
Step 6: Using the Standalone Forms Viewer Module

The Independent Forms Viewer module allows us to embed the form on a page of the BIZUIT Dashboard, ensuring direct access from the work environment.

Practical Example

1. We place the "**Standalone Forms Viewer**" module on a specific page of the Dashboard and configure it to load the standalone form we have created and enable versioning, ensuring that any updates are automatically reflected in the published version.

2. We verify that it is displayed correctly:



This guarantees that we will always be using the latest version of the form without the need for additional configurations.

Step 7: Testing and Final Validation

To ensure that the form works correctly, we will perform a series of tests:

- **We will verify the connection to the SQL database** and its correct response when querying customers.
- **We will confirm the execution of the BIZUIT process**, ensuring that each request generates an instance of the pending process in the corresponding activity.
- **We test access using the anonymous URL**, verifying that any user can interact with the form without the need for authentication.
- **We overhauled the integration in the Dashboard**, ensuring that the form is accessible from the Standalone Forms Viewer module.

In this demo, we set up a standalone form in BIZUIT from scratch, addressing advanced topics such as:

- Integration with SQL Data Sources, Invoking BIZUIT Internal Processes
- Implementation of validations and security
- Generation of anonymous access and invocation URLs and Integration in the Dashboard with support for versioning.

This case study demonstrates how standalone forms can be used to centralize key operations, improving the efficiency and accessibility of business processes.

Conclusion

In this unit, we explore how standalone forms allow you to collect, manage, and deploy information in BIZUIT without the need to integrate into a main process flow.

Thanks to their **independence**, these forms work autonomously and offer the flexibility to connect to various data sources (BIZUIT processes, REST APIs, and SQL statements). In addition, they make it easy to create **complex interfaces**, with advanced grids and controls, which can be shared via a URL and configured with different levels of authentication.

Key Points:

1. **Autonomy and Flexibility:** They do not depend on a main workflow, which allows them to be used in multiple scenarios (visitor registration, surveys, reporting dashboards, etc.).
2. **Connection with Data Sources:** REST APIs, SQL queries and BIZUIT processes can be integrated, centralizing information management.
3. **Access and Security:** Forms can be configured with anonymous access or custom authentication, and exposed using a unique URL or integrated into the Dashboard.
4. **Creation of Complex Interfaces:** They allow the design of screens with various controls, data grids and validations, facilitating the construction of advanced solutions for different equipment.
5. **Hands-on Demo:** We took a step-by-step look at how to create and configure a standalone form that validates data, invokes processes, and publishes to the Dashboard with versioning support.

Standalone forms in BIZUIT offer a versatile tool to address multiple data capture, visualization, and interaction needs within a corporate environment.

By allowing the combination of different sources of information and the independence of a main flow, they become an essential resource to streamline processes, offer personalized experiences and respond quickly to changing requirements.

With these fundamentals, we are prepared to delve into more advanced configurations and extend the use of BIZUIT in digital transformation scenarios.

Unit 4: Using Custom Modules Using Components

In this unit, we will explore the functionality of components in BIZUIT Dashboard and how they allow us to develop custom modules tailored to specific business needs programmatically. We will learn how to install, configure, and manage them to optimize the use of the platform.

Introduction to Components in BIZUIT Dashboard

In this section, we will discover how the component functionality expands the capabilities of the Dashboard, allowing us to develop modules that fit the specific processes and needs of each company.

Components not only add visual elements to the platform, but also allow us to implement advanced functionalities through traditional programming. We will learn what components are, how they can improve the BIZUIT experience, and how we can install, configure, and manage them in our Dashboard.

We will also explore case studies that will help us visualize your implementation in real-world scenarios, such as document validation, purchasing flow management, or customizing screens for our customer service team.

What is a Component in the Context of BIZUIT Dashboard?

A component in BIZUIT Dashboard is a programmable custom module that allows us to implement specific processes or screens within the platform, improving the user experience and optimizing business flows.

These modules can:

- Perform complex validations.
- Connect with external systems.
- Guide users through personalized steps within an interactive wizard.

Components are especially useful when we need to tailor the platform for specific use cases, thus improving the productivity and efficiency of our users.

Advantages of Components for Specific Business Cases

Implementing components in BIZUIT Dashboard offers us multiple benefits:

- **Programming Flexibility:** It allows us to develop any business logic through code, without predefined limitations.
- **Automation and Complex Validation:** We can set up modules that follow a structured flow, such as a wizard that guides users through several steps before validating a process.
- **Integration with External Systems:** Components can exchange data with other business systems automatically, without the need for the user to perform manual actions.
- **Personalized User Experience:** They allow us to create screens focused on specific tasks, optimizing times and reducing the operational burden.

Practical Usage Scenarios for Components

To better understand the impact of components in BIZUIT, let's discuss some examples in which these modules can transform business flows:

Validation of Academic Documentation: A component can be configured as a step wizard that guides the user through the process of verifying academic degrees. This module could include:

- Document upload.
- Automatic data validations.
- Approval or rejection based on specific rules.

Purchase Authorization Flow: In the purchasing area, a component can manage a multi-approval process, where different users with specific roles review and approve requests at various stages.

Claims Management Screen: For the customer service team, we can develop a component that facilitates claims management, allowing:

- Record and classify the type of claim.
- Automatically assign it to a specific area.
- Closely monitor the resolution process.

Hands-On Demonstration: Component Management

In the component management panel, we can see a list of all installed modules, along with details such as:

- Name of the component.
- Installed version.
- Author of the module.

Some examples of components available in BIZUIT include:

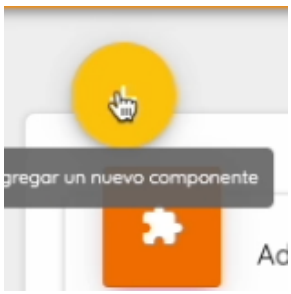
- Template Generator
- File Manager.

From this panel, we can also download components from one environment and reuse them in another.

Installation of New Components

To add a new component:

1. Click on the **"+"** button at the top left of the panel.



2. Select the corresponding file from the pop-up window.
3. We press "Install component" to complete the process.

We can consult the detailed information of each subcomponent and its functionality in the information icon.

Component Upgrade

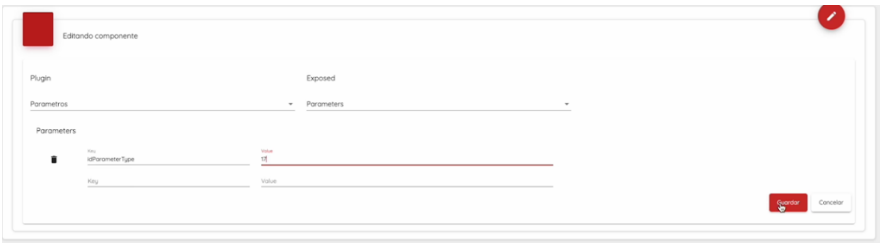
If we need to improve or fix bugs in an already installed component, we can easily update it: From the update icon, we replace the current component with its new version, keeping all its settings and data.

If a component is no longer needed: Click the trash can icon to delete it. A confirmation window will be displayed before proceeding, as the deletion is permanent.

Configuring Component Usage

Once a component is installed, we must configure it in the Dashboard so that it can be used correctly. To do this:

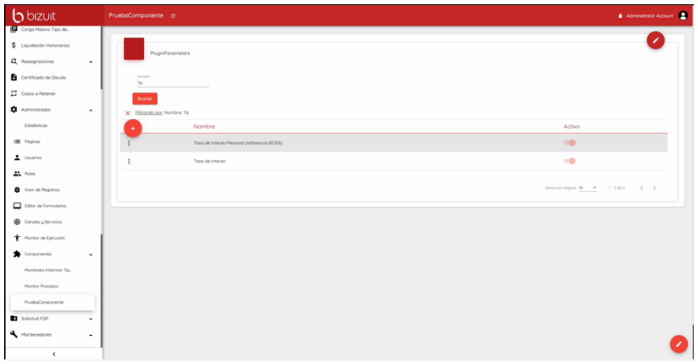
- We added a "Components" module to a page in the BIZUIT Dashboard.
- We edit the module and select the component we want to use from the drop-down menu.
- We configure the internal subcomponents, choosing which one we want to expose, such as: *About*, *Parameters*, *ParametersTypes*
- We define custom parameters, entering the keys and values necessary to adjust the behavior of the component.
- We save the changes and verify that the settings are correct.



Testing and Validation

Before finalizing, we perform tests on the Dashboard to ensure that the component is working properly:

1. We verify its correct display within the page.
2. We confirm that the configured parameters are applied appropriately.
3. We check the interaction with external systems, if the component has active connections.



Conclusion

In this unit, we explore how the components in BIZUIT Dashboard allow us to develop custom modules tailored to specific needs programmatically.

We saw that these modules can perform complex validations, integrate with external systems, and guide users through structured flows through interactive wizards.

In addition, we analyze its key benefits, such as scheduling flexibility, process automation, integration with other systems, and customization of the user experience.

We also review practical examples of use, such as validating academic documents, managing purchase authorizations, and managing claims in customer service.

Finally, we carried out a practical demonstration in which we learned how to install, update, delete and configure components in the BIZUIT Dashboard, ensuring their correct implementation and validation through functional tests.

Thanks to this unit, we now understand how components can optimize business processes and improve efficiency within the platform.

Unit 5: Visual and Functional Optimization of the Dashboard

In this section, we'll share a number of best practices to ensure that the BIZUIT Dashboard is intuitive, functional, and truly useful for users.

A well-structured design not only improves the user experience, but also makes the platform more efficient in the daily operation of the company.

Organizing Visual Elements

To improve the usability of the Dashboard, it is essential to organize the elements in a logical and consistent way. This involves:

- Arrange modules and widgets according to their priority and frequency of use.
- Group related items together for easy access and confusion.

Practical tip: Place the most used modules at the top of the Dashboard or in a left side column, as users usually scan the screen from top to bottom and from left to right. This way, key elements will always be visible and accessible.

Simplification of Navigation

A Dashboard should be intuitive and have clear and simple navigation. To do this:

- Menus and buttons should be well-labeled and organized by category.
- The navigation structure should be as flat as possible, reducing the number of clicks needed to access key features.

Tip: Avoid excessive menus or submenus, as they can overwhelm the user. Instead, prioritize simple navigation, where the main elements are accessible in one or two clicks.

Consistency in Visual Styles

Maintaining a consistent appearance across the interface helps users better orient themselves and interact with the system more intuitively. Best practice:

- Use a consistent set of colors and styles in buttons, icons, and text.
- Apply the same color for primary actions (example: send) and another for secondary actions (example: cancel).
- Avoid mixing multiple typographies or graphic styles that generate confusion.

A consistent visual design reduces the learning curve and allows users to navigate with greater confidence.

Optimization of User Information

It is essential that the Dashboard is not overloaded with information. Displaying too much data in one place can lead to distractions and make it difficult to make decisions.

Strategies to optimize information

- Prioritize key indicators, highlighting only the most relevant data for each user.
- Use custom modules and filters to display specific information based on the user's profile.

Practical example: A sales user should immediately visualize their key metrics, without interference from information from other areas.
Customizing the interface to each profile improves the experience and efficiency of the Dashboard.

Implementing User Feedback

An effective system must provide immediate responses to user actions. This builds trust and allows for better interaction with the platform.

Examples:

- If the user completes an action, the Dashboard should display a confirmation message.
- When submitting a form, a pop-up message, or a visual change should indicate that the action was successful.

This type of feedback helps users feel confident that the system has responded correctly to their interactions.

Conclusion

By applying these best practices, we make the BIZUIT Dashboard not only functional, but also easy to use and adapted to the daily needs of users.

Throughout this unit, we have learned how to:

- Configure and customize the Dashboard to optimize its performance.
- Apply design and navigation principles that enhance the user experience.
- Ensure accessibility and efficiency through a well-organized interface.

A well-optimized Dashboard facilitates decision-making, increases productivity, and generates a smoother and more positive experience for all users.

Chapter Summary

In this chapter we went through all the tools that BIZUIT offers to customize the Dashboard and turn it into a space **fully aligned with the identity, needs and processes** of each organization.

We started with the **configuration of colors, themes, logos, and icons**, which allowed us to capture the company's brand on the platform and improve the user experience with a visually cohesive environment.

We also learned how to **customize the login screen**, define the home page, and adjust the way forms are displayed, achieving a clear and easy-to-use interface.

Then, we delved into the **key modules** of the Dashboard: **Built-in Framework**, to display external content without leaving BIZUIT; **HTML Viewer**, which allows us to insert dynamic and customized content; and **Processes**, thanks to which users can start the most important tasks directly from the Dashboard.

Later, we learned about **independent forms**, a super versatile tool to capture and deploy data without being tied to a strict process flow, with the possibility of integrating with REST APIs, SQL statements and internal processes. We also saw how these forms can centralize complex actions and facilitate different business scenarios.

Continuing with advanced customization, we explore the **creation of components** (modules customized by code), which allow validations to be automated, interactive interfaces to be built, and BIZUIT to be connected to external systems in a deeper way.

Finally, we review **design** and usability best practices to make the Dashboard intuitive, attractive, and efficient: from organizing visuals and simplifying navigation, to consistency of styles and optimizing information.

In conclusion, this chapter gave us a complete overview of how to adapt the BIZUIT Dashboard to the specific requirements of our organization, both aesthetically and functionally.
By applying these tips and settings, we can **streamline workflows**, improve the **user experience,** and take full advantage **of the platform's flexibility to drive productivity and innovation.**