Chapter 13: Extending Functionality through Plugins

In this chapter, we will comprehensively address the **complete work cycle with a component in BIZUIT Dashboard**, from the initial idea to its implementation in production. We will start with a review of the key concepts of Unit 1 to understand what the components are and how, thanks to **traditional programming**, they are natively integrated with advanced functionalities of the platform: monitoring, traceability, security and unified presentation.

We will also explore **its strategic benefits**, such as process automation, real-time integration with external systems, and deep customization of interfaces.

These concepts will be illustrated with **real practical cases**, showing how the components turn the Dashboard into an adaptable tool with a high impact on the business operation.

The core of this chapter will be **Unit 2**, where we will break down the full lifecycle of a component:

- Technical development and interface design.
- Creation of installation package.
- Deployment in the Dashboard.
- Configuration adapted to business needs.
- Functional tests and adjustments before final use.

Ideal Audience

This chapter is designed for professionals who manage cloud, hybrid, or on-premise BIZUIT deployments, especially system administrators, database administrators, and support engineers.

Objectives

1. **Understand the concept and purpose of components in BIZUIT Dashboard:**

   Explore how custom components allow you to design processes and displays tailored to specific business needs.

2. **Learn the complete work cycle with a component:** From its development through traditional programming to its installation, configuration and testing in the Dashboard.
3. **Identify the key advantages of the components:** Analyze their flexibility, ability to integrate and improve the user experience.
4. **Apply concepts through practical examples:** Examine use cases, such as data validation, purchasing flows, and claims management.
5. **Acquire practical skills to customize the Dashboard:** Develop, configure, and test components in a real environment.

Unit 1: Fundamentals and Potential of Components

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.

What is a component in BIZUIT Dashboard?

A component is a programmable module designed to solve specific business needs. Thanks to its development with traditional programming (for example, Angular in the case of visual plugins), it is possible to:

- Implement complex processes.
- Design fully customized screens.
- Integrate seamlessly with the BIZUIT ecosystem.

Main strengths:

- **Native integration:** coherent operation within the ecosystem.
- **Centralized monitoring**: recording and monitoring of each action.
- **Traceability:** Auditing and documenting interactions, essential for regulatory compliance.
- **Advanced security:** access control, authentication, and data encryption.
- **Visual consistency:** unified aesthetics with the rest of the Dashboard.

Example: A company may require a dedicated interface to validate complex data connected to an external system, updating the information in real-time.

Key Advantages:

- **Flexibility:** Creating bespoke solutions that go beyond standard features.
- **Automation:** Reduction of manual work through automatic flows.
- **Integration:** connection with APIs and external platforms.

- **Optimized user experience:** intuitive interfaces adapted to each task.

Featured Case Studies

- **Validation of academic documentation:** guided flow with upload, automatic validation and approval/rejection according to rules.
- **Purchase authorization:** multiple stages with management of roles, permits and traceability.
- **Claims management:** registration, classification, assignment and follow-up with visual reports.
- **Internal training:** course catalog, registration and progress monitoring.
- **Marketing campaigns:** planning and control of KPIs with integration with Google Ads.

Conclusion

The components in BIZUIT Dashboard represent one of the most powerful tools to take business automation, integration, and personalization to an advanced level. Its programmable nature allows you to implement specific solutions that go far beyond standard functionalities, offering flexibility, scalability and a user experience adapted to each need.

Thanks to their native integration with the BIZUIT ecosystem, these modules inherit critical capabilities such as centralized monitoring, traceability, advanced security, and visual consistency, ensuring operational consistency and regulatory compliance. Key benefits—flexibility, automation, integration with external systems, and optimization of the user experience—make the components a strategic resource for solving complex scenarios and maximizing the value of the platform.

The case studies presented, from document validation to marketing campaign management, demonstrate that components not only cover specific operational needs, but also enable the digital transformation of entire processes, ensuring control, efficiency and adaptability in dynamic business contexts. In short, mastering its use is essential for those who seek to fully exploit the potential of the Dashboard and generate high-impact solutions.

Unit 2: Development, Installation, Configuration and Use of Components

In this unit we will focus on the complete cycle to take a component from its conception to its production within BIZUIT Dashboard. We will start with the clear definition of a business objective, we will advance through the technical development and interface design, and we will culminate with its packaging, installation, configuration and final validation.

The purpose is for us to be able to **create custom solutions that integrate business logic, data and user experience in a single module**, taking advantage of the flexibility provided by development with technologies such as Angular and native integration with BIZUIT.

Following this step-by-step process, we will not only learn how to produce functional and visually consistent components with the Dashboard, but also how to optimize them so that they respond to real needs, integrate with external systems, and comply with the platform's security, traceability, and monitoring standards.

This unit is based on the official training videos:

- https://www.youtube.com/watch?v=HgMd974AQqg
- https://www.youtube.com/watch?v=OfViJ0QNGMw
- https://www.youtube.com/watch?v=VXcFbwlUMe8

The content of these videos will serve as a practical guide to **transform ideas into components ready for productive use**, with an iterative approach that prioritizes quality, efficiency, and adaptability.

Before we dive into each stage, it's important to understand that the development of a component in BIZUIT Dashboard follows a **logical and structured flow** that allows us to move from an initial idea to a fully operational solution.

In the following sections, we will go through four key phases:

1. **Development of the component**, where we define its objective, structure the code, design the interface, connect the data and package it for installation.

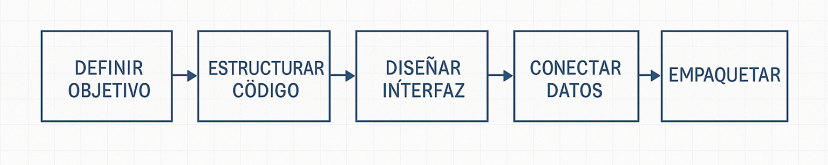
2. **Installation in the Dashboard**, ensuring that the component is properly integrated into the environment and ready to be used.
3. **Configuration of the component**, adapting it to the specific requirements of the business and taking advantage of its flexibility.
4. **Functional testing**, to validate that it serves its purpose and optimize its behavior before putting it into production.

Each of these steps is designed to **maximize the quality, efficiency, and consistency** of the solution within the BIZUIT ecosystem, ensuring that the end result not only works, but brings tangible value to our processes.

1. Component Development

At this stage we give shape to the initial idea, transforming it into a functional module. We clearly define the objective we seek to achieve, structure the code following best practices, design a visually coherent interface with the Dashboard, integrate the necessary data sources and prepare the final package ready for installation. This process is the technical heart of every component.

General Flow



Detail of each phase:

- **Define objective:**
  - Identify problem to be solved, process to automate or interface to customize.
  - Examples: data validation, integration with external APIs, approval flow optimization.
- **Structure Code:**
  - Create modules and components (Angular).
  - Implement services for communication with APIs.
  - Define data models.
- **Design interface:**
  - Use Angular Material or BIZUIT share libraries.
  - Ensure consistent aesthetics with the Dashboard.

- **Connect data:**
  - Integrate external sources via REST/GraphQL.
  - Validate, transform, and present data.
- **Pack:**
  - Configure Webpack and update file definition.
  - Prepare ZIP for installation in BIZUIT Dashboard.

Development Flow Diagram

2. Installation in BIZUIT Dashboard

Once the component is developed and packaged, we proceed to incorporate it into the BIZUIT Dashboard environment. Here we verify that the installation is successful, that the module appears correctly in the list of available components and that it has all the necessary information, such as version, author and status.

1- Access the **Modules/Components** → Admin Panel .
2- Upload the packaged ZIP file.
3- Confirm installation and verify:
   - Presence in the list of modules.
   - Version, author, and status data.

3. Component Configuration

With the component already installed, it is time to adapt it to the specific needs of the business. This involves assigning it to specific pages, adjusting operating parameters, defining

permissions, and, if applicable, configuring internal subcomponents. The right configuration is key to taking advantage of its full potential.

- Assign it to a Dashboard page.
- Define keys and values for personalization.
- Configure internal subcomponents if applicable.

4. Functional Testing

The last phase before putting the component into production is to validate that it meets its objective under real conditions. We test against different parameters, verify the integrity of the data and interaction with the rest of the Dashboard, and make iterative adjustments to ensure optimal performance and a satisfactory user experience.

- Validate that it meets the defined objective.
- Test behavior with different parameters.
- Adjust iteratively until optimal performance is achieved.

Conclusion

In this unit we learned how to:

- **Programming and packaging a component**, understanding its life cycle and applying best practices to ensure quality and maintainability.
- **Install it in the Dashboard**, integrating it in a coherent way with the rest of the platform and taking advantage of its native capabilities.
- **Configure and adjust the component** to respond to real needs, adapting parameters, permissions and behavior according to business requirements.
- **Validate it through functional testing**, ensuring that its performance and functionality meet expectations before going into production.

With this knowledge, we are in a position to transform BIZUIT Dashboard into a strategic tool fully aligned with our organization's objectives, capable of maximizing operational efficiency, strengthening process control and elevating the user experience. Mastering these steps not only allows us to respond to specific needs, but also to anticipate future challenges, scaling and evolving our solutions as the business grows.

Chapter Summary

Throughout this chapter, we first address **the fundamentals and potential of the components** in BIZUIT Dashboard, understanding that they are programmable modules designed to solve specific business needs. Its development with technologies such as Angular allows complex processes to be implemented, custom interfaces to be designed, real-time data to be integrated, and native capabilities such as centralized monitoring, traceability, advanced security, and visual coherence.

We identify the **key advantages** that make them a strategic resource: flexibility to create custom solutions, automation to reduce manual work, integration with APIs and external platforms, and optimization of the user experience. In addition, we reviewed **case studies** that showed their applicability in different contexts, from document validation and claims management to internal training and marketing campaigns.

In the second part, we focus on the **complete cycle of developing, installing, configuring, and validating** a component. We learned how to:

- **Develop it** by defining the objective, structuring the code, designing the interface, connecting data and packaging it.
- **Install it** in the Dashboard, verifying its correct incorporation and metadata.
- **Configure** parameters, permissions and behavior to adapt it to real needs.
- **Validate it** through functional tests, ensuring its performance and operation before going into production.

With this knowledge, we are in a position to turn BIZUIT Dashboard into a strategic tool fully aligned with the organization's objectives, capable of offering customized, scalable and high-impact solutions. Mastering both the conceptual vision and the technical process allows us not only to respond to current needs, but also to anticipate future challenges, evolving along with the demands of the business.