**Guía 3 Principios SOLID**

**Arquitectura de Software**

**Carlos Eduardo Mujica Reyes**

**Julián Mauricio Montoya Rodríguez**

**Bryan Andres Puentes Vargas**

**Universidad Manuela Beltrán**

**Bogotá D.C**

**2 de marzo de 2025**

**PRODUCT BACKLOG**

For this project, 2 methods of prioritizing the tasks that will be developed in the project will be used, the first is called Moscow and the second is called Business Value vs. Effort. The second one is not to hard to understand but we have to make a little explanation of how the Moscow method is used to prevent misunderstandings.

The MoSCoW method is a prioritization technique used in project management, business analysis, and software development. The acronym MoSCoW stands for four categories of prioritization:

* Must have: These are the essential characteristics without which the project would not be successful. They should represent no more than 60% of the total project effort.
* Should Have: They are important but not essential
* Could Have: They are desirable but not critical
* Won’t have: Features that will not be implemented, at least not at the current stage of the project

Now that we have explained about the Moscow method, we will begin to make the Product Backlog about some tasks we need to do for this ERP. We will not Use all the tasks of our project, but we will show plenty of them.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story** | **MOSCOW** | **Business Value** | **Effort** | **Final Priority** |
| US1 (Production) As an operator, I want to record production in real time to improve traceability | Must | 5 | 3 | 1.67 |
| US2 Production As a supervisor, I want to display production reports with details of each batch. | Must | 5 | 4 | 1.25 |
| US3 Production As a supervisor, I want to record quality defects in production. | Should | 4 | 3 | 1.33 |
| US4 Inventory As an operator, I want to manage the inventory of raw materials and finished goods. | Must | 5 | 4 | 1.25 |
| US5 Inventory As a user, I want to receive notifications when inventory is below a threshold. Should | Should | 3 | 4 | 0.75 |
| US6 Inventory As a manager, I want to view stock reports in real time. | Should | 4 | 3 | 1.33 |
| US7 Sales As a manager, I want to manage customer orders and their status to avoid delays. | Must | 4 | 3 | 1.33 |
| US8 Sales As a customer, I want to receive automatic confirmation when my order is processed. | Should | 4 | 3 | 1.33 |
| US9 Sales As a supervisor, I want a sales dashboard with key metrics. | Should | 5 | 4 | 1.25 |
| US10 Financial As an accountant, I want to generate financial and accounting reports from the ERP. | Should | 3 | 5 | 0.60 |
| US11 Financial As an administrator, I want to integrate the ERP with an electronic invoicing system. | Should | 4 | 3 | 1.33 |
| US12 (Financial As a manager, I want to visualize production cost analysis to make strategic decisions | Must | 5 | 4 | 1.25 |
| US13 (Human Resources) As an HR manager, I want to manage employee, shift, and attendance information. | Could | 4 | 3 | 1.33 |
| US14 (Human Resources) As a manager, I want to generate employee productivity reports | Should | 3 | 4 | 0.75 |
| US15 (Human Resources) As an employee, I want to receive notifications about my shifts and attendance. | Must | 5 | 4 | 1.25 |

**DEFINITION OF DONE**

A feature is considered "Done" when the code has been developed following quality norms or standards, storing that code on GitHub and continuously inspecting it. It must be implemented respecting previously established design patterns such as layer architecture or MVC.

On a continuous basis, it is required to do validation by doing unit tests that have at least 80% coverage in critical functions, and integration tests will also be used to ensure that the modules can communicate with each other without any errors or failures. The user interface must be responsive and must work on different devices, which must be considered for corrections before delivering the project. Each user story must meet its respective acceptance criteria and must be validated by the team. The information must be stored correctly in the databases and part of the documentation must be updated in the repository or in any other tool that meets what is necessary to store the relevant documentation or information on the project. Finally, the functionalities must always be available in a test environment and have their respective documentation, also take into account the approval by the client to make the necessary changes

**PLANNING POKER**

For the development of the first sprint, a user story will be selected for each module, then an approximate estimate of the effort required will be assigned using the poker planing with the fibonacci sequence.

**Selected User Stories**

1. Production

* User Story (US1): As an operator, I want to record the production in real time to improve traceability.
* Priority: High
* Effort Points (Planning Poker): 8 (Medium-high complexity, database integration and validations)

Acceptance Criteria:

* The system allows you to enter production data per batch.
* Errors or defects can be recorded in real time.
* The information is stored in the database and is accessible from the dashboard.
* It can be filtered by date and product type.
* Only authorized operators can perform registrations.

Necessary Tasks:

* Design the production registration interface.
* Implement the data entry form.
* Connect to the database to store records.
* Add validations to prevent incorrect data.
* Conduct functional tests and final adjustments.

1. Inventory

* User Story (US2): As an operator, I want to manage the inventory of raw materials and finished products.
* Priority: High
* Effort Points (Planning Poker): 5 (Medium complexity, requires interaction with multiple data)

Acceptance Criteria:

* Stocks of raw materials and finished goods can be added and updated.
* The system displays the stock in real time.
* Alerts are generated when a material is below the minimum threshold.
* The information is accessible only to users with permission.
* You can view the history of changes in inventory.

Necessary Tasks:

* Create the inventory table in the database.
* Design the inventory management interface.
* Schedule stock updates and alerts.
* Implement data validations.
* Test and document functionality.

1. Sales

* User Story (US3): As a manager, I want to manage customer orders and their status to avoid delays.
* Priority: High
* Effort Points (Planning Poker): 8 (High complexity, multiple states and business logic)

Acceptance Criteria:

* New orders can be registered in the system.
* The order status is updated (pending, in process, delivered).
* The manager can view the order history.
* A notification is sent to the customer when their order changes status.
* Only authorized users can modify orders.

1. Financial

* User Story (US4): As an accountant, I want to generate financial and accounting reports from the ERP.
* Priority: High
* Effort Points (Planning Poker): 13 (High complexity, advanced calculations and report generation).

Acceptance Criteria:

* Monthly and annual financial reports can be generated.
* Information is obtained in real time from the database.
* Reports can be exported to PDF or Excel.
* Only accountants can access this function.
* Custom filters can be applied (by date, type of transaction, client, etc.).

Required Tasks:

* Design the data structure for the reports.
* Implement report generation in the database.
* Create the user interface for visualization and export.
* Optimize query performance for large volumes of data.
* Perform tests with real data and document functionality.

1. Human Resources

* User Story (US5): As an HR manager, I want to manage employee information, shifts and attendance.
* Priority: High
* Effort Points (Planning Poker): 5 (Medium complexity, structured data handling).

Acceptance Criteria:

* New employees can be registered in the system.
* Shifts can be assigned and attendance can be displayed.
* Only HR managers can modify this information.
* An attendance report is generated for each employee.
* Employees can consult their attendance history.

|  |  |  |
| --- | --- | --- |
| **User Story** | **Effort Points** | **Reason Of Estimation** |
| Production | 8 | There is a complexity medium-high for validations and connections to the database |
| Inventory | 5 | There is an interaction with multiple data, but there isn’t a complex logic |
| Sales | 5 | It requires a Business logic |
| Financial | 13 | Advanced reports, financial choices and finally export of data |
| RRHH | 5 | There isn’t a complex structure, it just requires a structured management of employees |

**SPRINT TEST PLANING**

**Test Cases - Sprint 1**

**US1: Real-time production registration**

Enter valid data and verify that the batch is registered correctly.

Attempt to record without completing all required fields and validate that the system prevents it.

Record a batch with defects and verify that the system stores it correctly.

Attempt to access the function with an unauthorized user and validate the restriction.

Confirm that the data entered appears correctly on the dashboard.

**US2: Inventory management**

Add raw material and verify that the stock is updated.

Reduce stock and verify that the system reflects the change.

Simulate a reduction to activate the low stock alert and validate its operation.

Check that each inventory change is recorded in the history.

Attempt to modify the inventory with a user without permissions and validate that access is restricted.

US3: Order management

Create an order and verify that it is stored correctly.

Modify the order status and validate that it is updated in the system.

Change the status of an order and verify that the customer receives the corresponding notification.

Access the order history and validate that it is displayed correctly.

Attempt to modify an order with an unauthorized user and validate the restriction.

US4: Generation of financial reports

Generate a monthly report and verify that the data is accurate.

Apply date filters and validate that the results are correct.

Export a PDF report and check that it is generated without errors.

Download a report from Excel and validate that the format is as expected.

Attempt to generate reports with a user without permissions and validate the restriction.

US5: Employee and shift management

Register a new employee and validate that the data is stored correctly.

Assign a shift to an employee and verify that the information is updated.

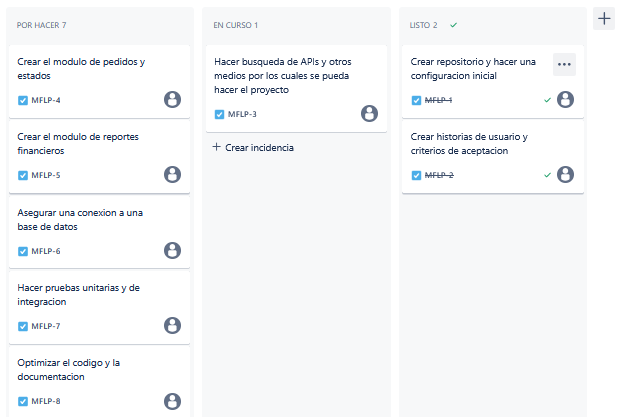
Consult an employee's attendance history and verify that the data is correct.

Generate an attendance report and validate its accuracy.

Attempt to modify employee information with a user without permission and verify the restriction.

|  |  |  |
| --- | --- | --- |
| Strategies | | Responsible |
| Technical aspects | Nothing complicated, we will make use of the necessary tools to fulfill the user stories. | Julian |
| Communication | Basic information channels will be used to let people know about things that need to be done or errors that occur in the development of the project. | Bryan |
| Quality assurance | Quality control will be performed every time a change or addition is made to the project to ensure that the best work is being done at all times. | Bryan |

**WORCK SCHEDULE**

****