

# Public Report

Simplified Management System for Microentrepreneurs (MEIs)

GitHub Repository: <https://github.com/orgs/Inteli-College/teams/2025-1a-t02-g57>

---

## Module 3 - System Development and MVP Preparation

### General Observations

This module aimed to transform the consolidated requirements from previous phases into a functional, modular, and integrated system, ensuring operational stability and adherence to defined technical specifications. Execution was divided into four biweekly sprints, covering technical planning, backend and frontend construction, component integration, and MVP launch preparation.

The focus was to ensure architectural coherence, scalability, technical documentation, and support for the go-to-market strategy outlined in the previous module.

---

## Sprint 1 (Weeks 1–2): Module Planning

### Planned Objectives

- Convert project requirements into a detailed technical plan.
- Define system architecture and key integrations.
- Map deliverables and artifacts for backend and frontend.
- Structure tools to support the MVP and go-to-market strategy.

### Activities Performed

- Creation of a technical plan consolidating architecture, technologies, and dependencies.
- Structuring of the development schedule across four sprints.

- Definition of modular architecture, segmenting backend and frontend.
- Documentation of MVP support tools.

## Adjustments from Original Plan

- Inclusion of additional artifacts to support the MVP, incorporated after system deployment studies.

## Generated Artifacts

### Project Plan

- **Strategic Conclusion:**
  - Aligned technical development with the commercial strategy, ensuring coherence between the product and target market.
- **Value Roadmap:**
  - Guided future deliveries toward market validation and modular expansion.

### Observations:

This sprint consolidated the transition from the conceptual phase to technical execution, laying the groundwork for structured system development.

---

## Sprint 2 (Weeks 3–4): Backend Construction

### Planned Objectives

- Develop backend logical architecture, separating responsibilities between Server (back-office) and POS (point of sale).
- Implement internal communication structure for the POS using Electron's IPC pattern.
- Build controllers and models for business logic and MySQL data persistence.
- Define and document directory structure and main communication channels.
- Create User Stories for backend development of the Server.

### Activities Performed

- Definition of a dual backend architecture: a centralized Server for operational support and a local POS as the MVP's main interface.
- Implementation of internal POS architecture with a modified MVC pattern using Electron, TypeScript, Node.js, and MySQL.
- Structuring communication between the main (backend) and renderer (frontend) processes via ipcMain and ipcRenderer, with preload.ts acting as a secure bridge.
- Development of controllers for business rules, models for database access, and connection.ts for connection management.
- Creation of Server backend User Stories focusing on registration and management flows for clients, users, and workspaces.

## Adjustments from Original Plan

- Creation of Server User Stories was formalized as an additional scope delivery, addressing identified go-to-market needs.

## Generated Artifacts

### POS Backend Source Code

- **Final Observations:**
  - The architecture ensures modularity, security, and scalability, with a clear separation between business logic, data access, and communication interface.
- **Conclusion:**
  - Backend structure completed, providing a robust base for data persistence and system operations, ensuring secure hardware communication via serialport.

### Server User Stories Document

- **Strategic Conclusion:**
  - Defines functional requirements for the back-office component, enabling centralized support for operations and future management of multiple POS units.

## **Observations:**

This sprint solidified the system's backbone, ensuring business logic and data persistence were implemented in an organized, scalable manner to sustain MVP functionalities.

---

## **Sprint 3 (Weeks 5–6): Frontend Construction**

### **Planned Objectives**

- Develop POS frontend focused on user experience.
- Implement main point-of-sale functionalities.
- Structure code and components in a modular, scalable way.
- Integrate build, linting, and CI tools.

### **Activities Performed**

- Implementation of registration, sales, reports, and settings screens.
- Project structured in modular folders (assets, components, pages, hooks, etc.).
- Integration of technologies: React, TypeScript, Tailwind CSS, Vite, Context API, and GitHub Actions.
- Documentation of architecture and development practices.

### **Adjustments from Original Plan**

- No structural adjustments — architecture maintained as planned.

### **Generated Artifacts**

#### **POS Frontend – Code Structure and Interface**

- **Final Observations:**
  - Application organized following best practices for componentization and code standardization.
- **Technical Conclusion:**

- Frontend reached sufficient technical maturity for integration and testing with the backend.

### **MVP Demonstration (Frontend)**

- **Strategic Conclusion:**

- Demonstration validated interface navigation and usability, even in environments disconnected from the backend.

- **Value Roadmap:**

- Provides the visual and functional foundation for final system integration.

### **Observations:**

This sprint consolidated the frontend as the main MVP interface, prioritizing visual clarity and technical usability.

---

## **Sprint 4 (Weeks 7–8): System Integration and MVP Testing**

### **Planned Objectives**

- Integrate backend and frontend in a functional environment.
- Validate internal communications and performance.
- Execute initial stability and usability tests.
- Consolidate functional MVP version.

### **Activities Performed**

- Integration of ViaFacilis system components.
- Installation, execution, and usage flow tests.
- Communication tests between modules and validation of user stories.
- Documentation of functionalities and test results.

### **Adjustments from Original Plan**

- No significant adjustments reported.

## Generated Artifacts

### ViaFacilis Integrated System (v1.0.2)

- **Final Observations:**
  - Functional system with full backend–frontend integration.
- **Technical Conclusion:**
  - Integration confirmed system stability and compliance with technical specifications.

### MVP Testing and Demonstration Report

- **Strategic Conclusion:**
  - MVP met minimum functionality and usability criteria, ready for homologation.
- **Value Roadmap:**
  - Start of operational support phase and tool structuring for launch.

### Observations:

This sprint finalized the functional MVP, closing the technical development phase and opening the path for launch preparation.

---

## Sprint 5 (Weeks 9–10): Launch Preparation

### Planned Objectives

- Structure tools and support materials for the MVP.
- Create a public report consolidating module deliveries.
- Document real user testing and collect feedback.
- Prepare homologation environment and launch materials.

### Activities Performed

- Preparation of the Public Report summarizing technical and strategic module outcomes.

- Execution and documentation of two user tests validating usability, navigation clarity, and stability of key POS functions.
- Recording user observations, difficulties, and suggestions from testing sessions.
- Implementation of minor adjustments based on collected feedback.

## **Adjustments from Original Plan**

- Formal inclusion of user test documentation as part of MVP homologation artifacts.

## **Generated Artifacts**

### **Module Public Report**

- **Final Observations:**
  - Consolidated document summarizing the development cycle, results, and lessons learned.
- **Technical Conclusion:**
  - Ensures institutional transparency and standardized communication about project progress.

### **User Testing Reports (2 Sessions)**

- **Technical Conclusion:**
  - Tests confirmed system stability and overall usability, with minor interface and navigation flow adjustments applied.
- **Value Roadmap:**
  - User feedback will be incorporated into future continuous improvement and functional expansion iterations.

### **Observations:**

This sprint concluded the development cycle with the creation of the public report and direct user validation, closing the technical phase and initiating the homologation and post-launch monitoring phase.