



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING

Software Requirements Specification

Midterm Project Report

Course: Software Engineering I

Program: Computer Engineering

Instructor: Mgtr. Vanessa Jurado Vite

Team Members:

- Hilda Victoria Angulo Borja
- Fabricio Alberto Chang Encalada
- Alejandro Francisco Sornoza Vera

Contents

1	Introduction	4
2	Problem Definition	4
3	General Objective	4
3.1	Specific Objectives	4
4	Project Scope	5
5	Selected Process Model – SCRUM Methodology	5
5.1	Roles	5
6	User Stories	6
6.1	Sprints	6
7	System Requirements	6
7.1	Functional Requirements	7
7.2	Non-Functional Requirements	9
8	Activity on Arrow	11
8.1	Task List for System Implementation	12
8.2	Activity on Arrow Diagram	13
9	Scheduling	13
9.1	Scheduling table	14
9.2	Time-Limited Scheduling	15
9.3	Resource-Limited Scheduling	16
10	Prototype	16
10.1	Page 1: Success	16
10.2	Page 2: Failure	17
10.3	Page 3: TV	19
10.4	Page 4: Administrator	19
10.5	Page 5: Web User	20
11	Act of acceptance	21
12	References	23
13	Annexes	24
13.1	Annex 1: Excel sheet with user stories	24
13.2	Annex 2: User Stories displayed on Jira	33
13.3	Annex 3: Prototype flows from Figma	51

List of Figures

1	AoA Diagram showing task dependencies and event times	13
2	Time-Limited Scheduling (TLS) diagram — smoothed for balanced resource usage.	15
3	Resource-Limited Scheduling (RLS) diagram — smoothed to reflect realistic resource constraints.	16
4	Prototype - Page 1: Success of the touch kiosk.	17
5	Prototype - Page 2/Flow 1: Empty fields.	18
6	Prototype - Page 2/Flow 2: Invalid information.	18
7	Prototype - Page 3: TV	19
8	Prototype - Page 4: Administrator	20
9	Prototype - Page 5: Web User QR	20

List of Tables

1	Sprint breakdown for the system implementation plan	6
1	Functional Requirements of the System	7
2	Non-Functional Requirements of the System	9
4	Product Requirements classified under Sommerville	10
5	Organisational Requirements classified under Sommerville	10
6	External Requirements classified under Sommerville	11
7	Object-Oriented Task List for System Implementation	12
8	Scheduling table with duration, resources and timing constraints	14

1. Introduction

This report documents the development of a queue management system designed for use on a touchscreen kiosk, conducted by the SACODE team as part of an academic project. The main objective of the system is to allow users to request service tickets quickly, intuitively, and efficiently, thus improving service delivery in institutions or service centers with high customer flow.

To ensure an agile, collaborative, and user-centered development process, the Scrum methodology was applied to the project. Through iterative and incremental cycles, key functionalities were identified, prioritized, and developed based on well-defined user stories and acceptance criteria.

This report presents the work completed from the initial planning stages to the deliverables achieved in the first project phase, including the definition of requirements, the organization of backlog, the execution of the sprint, and the results of each iteration.

2. Problem Definition

In the Ecuadorian market, existing queue management systems are often designed for specific use cases, offering limited options for customization and adaptability. This rigidity prevents small businesses, diverse organizations, or institutions with unique workflows from implementing a solution that truly fits their operational needs.

The client seeks a queue management system that not only automates the ticketing process, but also provides a flexible, customizable, and accessible tool for all types of users and sectors. The goal is to create a differentiation solution that allows companies, institutions, or entrepreneurs to tailor the system to their way of working -defining their own service types, interface, priorities, and operational logic without being constrained by generic or closed systems.

This project aims to design and develop an innovative queue management system that, through a user-friendly interface and advanced configuration options, becomes a versatile and scalable alternative to the traditional systems currently available in the country.

3. General Objective

Develop an adaptable, customizable, and user-friendly queue management system that enables all types of users, such as businesses, entrepreneurs, or institutions, to efficiently manage customer service through an intuitive interface, a configurable administration panel, and accessible display mechanisms, thereby standing out from existing systems due to its versatility, scalability, and cross-platform orientation.

3.1 Specific Objectives

- Design a modular architecture that allows the queue management system to be adapted to diverse types of business or institution without rewriting its core code.
- Implement an intuitive graphical interface for kiosks and service screens to ease end-user interaction.
- Develop an administration panel to configure services, types of attention, schedules, and priorities according to the client's needs.
- Incorporate real-time display features via TVs or mobile devices using QR codes.
- Ensure that the system works both online and offline to ensure service continuity.
- Validate the system with real users from different profiles to ensure usability and adaptability.
- Facilitate the installation and deployment of the system in diverse environments, including small businesses with limited technological infrastructure.

4. Project Scope

This project involves the design, development, and implementation of an adaptable and customizable queue management system, composed of four main components.

- **Touchscreen kiosk interface:** Allows users to autonomously generate a ticket through an intuitive interface.
- **Display TV:** Shows real-time information about current and upcoming events, as well as relevant messages.
- **Administrative panel:** Enables staff to configure the system according to their needs, including types of service, priorities, schedules, and monitoring the flow of attention.
- **QR-accessible web page:** Provides users with a digital way to track their ticket, showing estimated wait time, queue position, and real-time updates.

The system is designed to be used in various settings, such as public institutions, medical centers, businesses, private offices, and entrepreneurial ventures, adapting to each without requiring structural changes. Priority will be given to smooth user experience, clear interface design, and offline functionality in key components such as the kiosk. Ticket generation will be strictly on-site via the touchscreen kiosk, while remote access will be limited to viewing ticket status through a QR link printed on the receipt. Integration with external systems (e.g., CRMs or ERPs) and native mobile app development are not included in this phase.

5. Selected Process Model – SCRUM Methodology

For the development of the queue management system, the Scrum framework was selected as the process model. Given the nature of this project developed by a team of three students, in close collaboration with the client, and requiring incremental validation—Scrum provides the necessary flexibility and structure.

The project was managed through a series of sprint cycles, each including the main activities defined by the framework: Sprint Planning, Daily Scrum, Sprint Execution, Sprint Review, Sprint Retrospective, and Product Backlog Grooming. These activities structured the development rhythm and enabled the team to inspect and adapt at each iteration.

Throughout the sprints, user stories were selected and implemented progressively, with each cycle producing a potentially shippable increment. General details about the cycle and team coordination are provided in this section, while specific information about functional and non-functional requirements, system modules, backlog prioritization and sprints is addressed in the following sections of the document.

5.1 Roles

In our team, composed of three members, the SCRUM roles are distributed in this way:

Product Owner: Represented by our client, Juan Carrillo, who defines the system requirements, provides feedback on deliverables, and prioritizes the Product Backlog based on their needs. Communication is maintained through scheduled review sessions and approval checkpoints after each sprint.

Scrum Master: This role is assigned to Alejandro Sornoza, who is responsible for organizing SCRUM events, ensuring team focus, and removing any obstacles that may impact sprint progress. He acts as the facilitator between the development team and the Product Owner.

Development Team: Composed of Hilda Angulo and Fabricio Chang. Both actively collaborate on the design, implementation, and testing of the system components, including the kiosk interface, turn display screen, administrative dashboard, and QR tracking module. Tasks were distributed based on technical expertise and balanced workload.

Regarding project stakeholders, the institution's administrative client acts as Product Owner. Secondary stakeholders include system end-users (citizens or students), operators of the ticketing module, and future IT personnel responsible for system support. Their needs have been captured through research and reflected in user stories.

6. User Stories

In accordance with agile development principles, the functional requirements of the system were captured and refined using user stories. This approach allowed us to focus on the needs and expectations of different types of users in a clear and actionable way.

Each user story was written from the perspective of a specific user role and follows the standard format: *As a [type of user], I want [target], so [reason]*.

The user stories helped the development team define the system's functionalities iteratively, and they served as the foundation for sprint planning during the development cycle.

A summary of the main user stories defined for the system is presented in the annexes (see Annex 1). This file includes key fields such as user type, story description, acceptance criteria, priority level, and any observations or refinements made after feedback.

6.1 Sprints

The following table outlines the sprint breakdown for the system implementation phase. Each sprint was structured based on the project's task scheduling (Resource-Limited Scheduling) and aligned with user stories to ensure cohesive development. Sprint 0 covers all foundational technical tasks necessary to support the system, while Sprints 1 through 5 progressively deliver user-centered features, integrations, and final deployment. Sprint durations were adapted according to workload and task complexity.

Sprint	Duration (weeks)	Main Goal
Sprint 0	7	Technical setup and infrastructure (UI prototyping, Git setup, database schema, CI/CD pipeline, encryption algorithm, and OTP logic for first login).
Sprint 1	3	Entity modeling and basic system components (categories, kiosks, screens, branches, admin users, and advertisement content).
Sprint 2	3	Entity relationships and business logic setup (assignment of entities, user-role permissions, and priority rules).
Sprint 3	3	Authentication and user interaction (admin registration, turn generation, QR interaction, screen display, and profile editing).
Sprint 4	3	Notifications, error handling, and reporting (kiosk data setup, turn reinsertion, reporting, notifications, priority, and recovery).
Sprint 5	5	Final testing, validation, and system deployment (unit tests, integration tests, UAT, and production deployment).

Table 1: Sprint breakdown for the system implementation plan

7. System Requirements

This section defines the system requirements that guide the development and validation of the proposed solution. The requirements were elicited based on stakeholder interviews, domain analysis, and system goals, and are categorized into two main groups: functional and non-functional requirements.

Functional requirements describe the specific features and behaviors that the system must support to fulfill user and business needs. These include user interactions, system responses, and rules for data processing and workflow execution. Functional requirements are typically expressed through use cases and user stories that form the basis for sprint planning and implementation.

Non-functional requirements, on the other hand, define quality attributes and constraints that the system must satisfy. To organize these requirements in a systematic and industry-accepted manner, we adopted the classification proposed by Ian Sommerville. His model groups non-functional requirements into product requirements (such as performance, reliability, and usability), organizational requirements (like development

standards and deployment platforms), and external requirements (including regulatory, legal, and interoperability constraints). This classification ensures that all critical system qualities are addressed consistently throughout the system lifecycle.

By defining both functional and non-functional requirements clearly, we ensure traceability between user needs, system features, and technical implementation. The following tables present these requirements in structured formats to support planning, validation, and future maintenance.

7.1 Functional Requirements

ID	Description
FR1	The administrator must be able to receive a temporary password (OTP) by email in order to log in for the first time.
FR2	The administrator must be able to use the received OTP to create a password, complete personal data and access the system.
FR3	The administrator must be able to create new branches in the system.
FR4	The administrator must be able to create new kiosks in the system.
FR5	The administrator must be able to create new display screens for queue and advertising content.
FR6	The administrator must be able to create advertising content for screens.
FR7	The administrator must be able to create accounts for administrative users.
FR8	The administrator must be able to assign kiosks to registered branches.
FR9	The administrator must be able to assign display screens to branches.
FR10	The administrator must be able to assign service categories to kiosks.
FR11	The administrator must be able to assign administrative users to branches.
FR12	The administrator must be able to configure the required personal data to be collected at kiosks.
FR13	The administrator must be able to configure priority rules between categories and turn recovery.
FR14	The administrator must be able to manually reinsert a missed turn into the queue.
FR15	The administrator must be able to generate and export custom reports.
FR16	The client must be able to generate a turn from the kiosk.
FR17	The client must be able to check the status of their turn via QR code.
FR18	The client must be able to see on display screens when their turn is being called.
FR19	The client must be able to easily understand which category to choose at the kiosk.
FR20	The client must be able to view the estimated waiting time on display screens.
FR21	The client must be able to receive web notifications when their turn is approaching.
FR22	The client must be able to automatically recover a turn within the allowed time.
FR23	The vulnerable client must be able to receive priority service without being placed first or last.
FR24	The client who generates a turn must be able to receive a printed receipt with a QR code and tracking link.
FR25	The administrator must be able to modify their personal information during initial system access.
FR26	The administrator must be able to create new branches within the system.
FR27	The administrator must be able to create service categories for each branch.

FR28	The administrator must be able to create new kiosks in the system.
FR29	The administrator must be able to create new display screens for service and advertising.
FR30	The administrator must be able to create multimedia advertising content for screens.
FR31	The administrator must be able to create accounts for administrative users.
FR32	The administrator must be able to assign administrative users to specific branches.
FR33	The administrator must be able to assign kiosks and screens to branches.
FR34	The administrator must be able to assign categories to kiosks.
FR35	The administrator must be able to define which personal data are required at each kiosk.
FR36	The administrator must be able to define priority rules and turn recovery logic for each branch.
FR37	The administrator must be able to manually reinsert a lost or expired turn back into the queue.
FR38	The administrator must be able to generate and export customized reports by branch and service.
FR39	The client must be able to generate a service turn from the kiosk.
FR40	The client must be able to scan a QR code to check the status of their turn.
FR41	The client must be able to view on screens when their turn is being called.
FR42	The client must be able to understand easily which service category to choose.
FR43	The client must be able to see the estimated waiting time before being attended.
FR44	The client must receive web notifications when their turn is approaching.
FR45	The client must be able to recover a turn automatically within the configured grace period.
FR46	A vulnerable client must be given priority without being placed strictly first or last.
FR47	The client must receive a printed ticket with QR and tracking link upon generating a turn.

Table 2. Functional Requirements of the System

7.2 Non-Functional Requirements

ID	Description
NFR1	The OTP code must automatically expire after 5 minutes for security reasons.
NFR2	The system must ensure 99.9% availability.
NFR3	Passwords must be stored using secure hashing algorithms such as bcrypt.
NFR4	The system must use HTTPS to secure all communications between clients, kiosks, and servers.
NFR5	The kiosk interface must be optimized for touch screens and have response times under 2 seconds.
NFR6	The system must support at least 100 concurrent kiosks without performance degradation.
NFR7	The database must perform automatic backups every 24 hours.
NFR8	Configured priority rules must execute in less than 1 second.
NFR9	Advertising content must load in under 3 seconds on screens.
NFR10	The system must send web push notifications without requiring browser plugins.
NFR11	The system must display real-time turn status and updates on screens without page reload.
NFR12	The system must be modular and allow new features to be added without impacting existing functionality.
NFR13	Administrative operations must be protected by multi-factor authentication.
NFR14	The kiosk must display clear and user-friendly error messages when issues occur.
NFR15	Kiosk printers must be networked or locally configured to allow instant printing.
NFR16	Reports must be exportable in PDF and CSV format within 10 seconds.
NFR17	Customer personal data must comply with data privacy and protection regulations.
NFR18	The kiosk interface must provide accessibility features for users with disabilities (e.g., high contrast or screen reading).
NFR19	Display screens must operate in read-only mode without exposing system controls.
NFR20	The platform must include real-time monitoring to detect system failures or saturation.
NFR21	The system must validate the integrity of branch and user assignment data to avoid conflicts.
NFR22	Screen content must support video and image formats, loading in under 2 seconds.
NFR23	Reports must be exportable in PDF and Excel and generated within 10 seconds.
NFR24	The system must ensure QR code scans return real-time status with updates under 1 second.
NFR25	The kiosk interface must display category options in clear language and graphical icons.
NFR26	The system must allow reinserting turns without compromising queue fairness logic.
NFR27	Recovery logic must work offline on the kiosk and sync once reconnected.
NFR28	All screen messages must support accessibility standards (WCAG 2.1) including high contrast.
NFR29	Web push notifications must be compliant with modern browser APIs.
NFR30	Priority logic must allow exceptions while maintaining queue order for fairness.

Table 3. Non-Functional Requirements of the System

Non-Functional Requirements Sommerville Classification

Product Requirements

ID	Category	Description	Validation Criteria
NFR1	Security	The OTP code must automatically expire after 5 minutes for security reasons.	A unit test verifies OTP expires after 5 minutes and cannot be reused.
NFR2	Dependability (Availability)	The system must ensure 99.9% availability.	Availability will be tested with monitoring tools over a simulated period.
NFR3	Security	Passwords must be stored using secure hashing algorithms such as bcrypt.	A code inspection will verify bcrypt is used for hashing.
NFR4	Security	The system must use HTTPS to secure all communications between clients, kiosks, and servers.	Penetration testing will confirm HTTPS is enforced site-wide.

Table 4: Product Requirements classified under Sommerville

Organisational Requirements

ID	Category	Description	Validation Criteria
NFR14	Usability / Operational	The kiosk must display clear and user-friendly error messages when issues occur.	Usability tests will check clarity and readability of error messages.
NFR15	Operational	Kiosk printers must be networked or locally configured to allow instant printing.	Operational test will verify printing speed under different network settings.
NFR25	Usability / Operational	The kiosk interface must display category options in clear language and graphical icons.	User acceptance tests will confirm icon clarity and language readability.

Table 5: Organisational Requirements classified under Sommerville

External Requirements

ID	Category	Description	Validation Criteria
NFR13	Security (Regulatory)	Administrative operations must be protected by multi-factor authentication.	A security test will verify that MFA is required for all admin tasks.
NFR17	Regulatory (Privacy)	Customer personal data must comply with data privacy and protection regulations.	Legal audit will confirm compliance with applicable data privacy laws.
NFR18	Accessibility	The kiosk interface must provide accessibility features for users with disabilities (e.g., high contrast or screen reading).	Accessibility testing will confirm compliance with WCAG standards.
NFR28	Accessibility	All screen messages must support accessibility standards (WCAG 2.1) including high contrast.	Automated and manual tests will validate WCAG 2.1 conformance.
NFR29	Interoperability / Regulatory	Web push notifications must be compliant with modern browser APIs.	Browser compatibility tests will verify API compliance across platforms.

Table 6: External Requirements classified under Sommerville

8. Activity on Arrow

Task Planning Approach Based on Object-Oriented Design

In the development of this project, we are considering the use of **Object-Oriented Programming (OOP)** as the primary paradigm for designing and implementing the system. This decision directly influenced the logical sequencing of tasks reflected in our Activity on Arrow (AOA) diagram and task list.

Unlike a strict user-centered order—which is the natural structure of user stories (US)—the task list below has been restructured to follow a *bottom-up implementation strategy* consistent with OOP principles. In OOP, it is common and necessary to begin with the development of the most fundamental classes and objects that serve as building blocks for more complex components. For instance, since display screens are composed of advertising content, it would be illogical to implement screen-related functionalities before ensuring the advertising content module is in place.

This means that entities such as “advertising content,” “branches,” and “categories” are created before higher-level elements like “kiosks,” “screens,” or “assignments.” Moreover, user interface functionalities that rely on these base entities are scheduled for implementation later in the development process, once their dependencies are resolved. This approach not only reflects a technically coherent class hierarchy, but also facilitates modular testing, clearer dependencies, and improved maintainability.

In summary, the order of tasks presented in the following table prioritizes class dependency resolution over chronological user interactions, which aligns with the nature of a scalable and modular object-oriented system.

In accordance with the academic structure of the course, the total hours allocated for project development are divided into two distinct phases. During the current semester, the first six weeks (approximately 1.5 months) are designated for autonomous project work at a rate of 2 hours per week per developer. This limited time is primarily intended for foundational activities such as prototyping, database design, and system architecture planning. Following this period, the development work will continue into the next academic semester, which spans approximately 16 weeks (4 months). During this second phase, students are expected to dedicate 5 hours per week per developer, allowing for more sustained and focused implementation of system features, testing, and deployment. This planned distribution ensures a progressive workload that

8.1 Task List for System Implementation

ID	Description	Duration (hrs)	Estimated Weeks	Dependencies	Resources
1	Prototyping user interfaces	8	2	-	2
2	Setup version control (Git)	2	1	-	1
3	Design database schema	12	3	1,2	3
4	Initialize CI/CD pipeline	4	1	3	2
5	Define password encryption algorithm	4	1	2	2
6	Implement Category class/entity	5	1	4	1
7	Implement Advertisment class/entity	5	1	4	1
8	Implement Screen class/entity	5	1	7	1
9	Implement Kiosk class/entity	5	1	6,7	1
10	Implement Branch class/entity	5	1	8,9	1
11	Implement Permission class/entity	5	1	8,9	1
12	Implement Admin User class/entity	5	1	10,11	1
13	Implement Assignment functions	5	1	12	1
14	Implement Turn management logic	15	1	13	3
15	Implement Priority rules engine	10	1	14	2
16	Implement OTP logic for first login	5	1	5,12	1
17	Implement User profile setup/editing	5	1	16	1
18	Implement Authentication and Authorization	10	1	17	2
19	Turn request UI and QR receipt	15	1	15	3
20	QR scan and Turn tracking	10	1	19	2
21	Display turns on screen	5	1	20	1
22	Time estimation and Web notifications	10	1	20	2
23	Permissions management for admins	5	1	18	1
24	Admin dashboard (CRUD)	10	1	23	2
25	Report generation and export	10	1	24	2
26	Error reporting system	5	1	24	1
27	Unit testing for core modules	15	1	21,22,25,26	3
28	System and integration testing	15	1	27	3
29	Deploy to test environment	5	1	28	1
30	UAT and environment validation	5	1	29	1
31	Deploy to production	5	1	30	1

Table 7: Object-Oriented Task List for System Implementation

8.2 Activity on Arrow Diagram

The Activity-on-Arrow (AoA) diagram was developed by applying the concepts and methodologies learned during the course, particularly those related to project scheduling and dependency analysis. Each task was carefully analyzed to identify its prerequisites and logical sequence within the system's object-oriented development context. The resulting diagram visually represents the workflow and interdependencies, ensuring efficient time management and resource allocation throughout the project's execution. This structured approach allowed us to determine the earliest and latest event times, supporting a clear understanding of the project's critical path.

Additionally, when establishing task dependencies, care was taken to avoid redundancies by not including tasks that were already indirectly dependent through other prerequisite tasks. This helped to keep the dependency structure clean and minimized unnecessary complexity in the AoA diagram, ensuring clarity and logical consistency in the sequencing of project activities.

Figure 1 shows the AoA diagram developed for the project, where the dependencies between tasks are represented through directional arrows, and the earliest and latest event times—expressed in weeks—are labeled at each node.

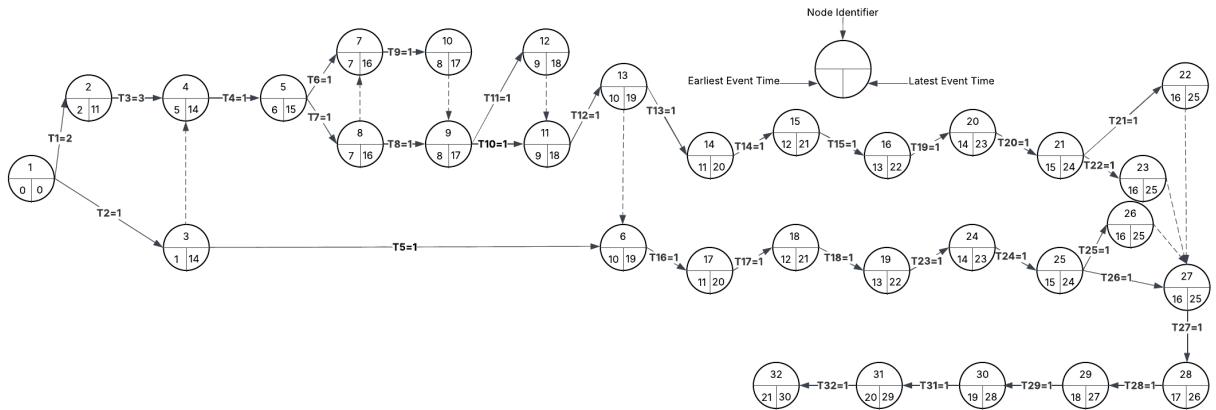


Figure 1: AoA Diagram showing task dependencies and event times

9. Scheduling

In this section, a detailed scheduling plan was created by applying the principles discussed in class, particularly those from the topics of time-limited and resource-limited scheduling. The process involved determining the earliest start times, resource assignments, task durations, and available float for each activity. By analyzing this data, we ensured that critical tasks were prioritized and resource usage remained efficient. This schedule provides a clear roadmap for executing the project within the desired timeframe, while identifying potential areas for optimization and parallel execution where applicable.

The delivery deadline for the project is estimated to be around January 9, 2026, which corresponds to 28 weeks from the starting point. Since steps 1 and 2 (which required two weeks) have already been completed, the full project timeline was planned for a total of 30 weeks. However, according to the Activity-on-Arc (AoA) analysis under a time-limited scheduling approach, all project tasks could potentially be completed in just 21 weeks. This results in a global float of 9 weeks at the project level, offering flexibility in case of delays or unexpected events.

Nevertheless, due to the course constraint of working with a maximum of three developers and without the possibility of external assistance, we adopted a resource-limited scheduling strategy. This approach considers the availability and capacity of team members, and was implemented to produce a feasible and sustainable execution plan under limited human resources. The final schedule reflects this balance between theoretical efficiency and practical constraints.

9.1 Scheduling table

The following table summarizes the scheduling details for each activity, including the required staff, estimated duration, earliest start time, float, and latest finish time. This information was derived from the Activity-on-Arrow analysis and helps ensure that project execution aligns with resource availability and timing constraints.

ID	Description	Staff	Duration	Earliest Start	Float	Latest Finish
1	Prototyping user interfaces	2	2	0	9	11
2	Setup version control (Git)	1	1	0	13	14
3	Design database schema	3	3	2	9	14
4	Initialize CI/CD pipeline	2	1	5	9	15
5	Define password encryption algorithm	2	1	1	17	19
6	Implement Category class/entity	1	1	6	9	16
7	Implement Advertisment class/entity	1	1	6	9	16
8	Implement Screen class/entity	1	1	7	9	17
9	Implement Kiosk class/entity	1	1	7	9	17
10	Implement Branch class/entity	1	1	8	9	18
11	Implement Permission class/entity	1	1	8	9	18
12	Implement Admin User class/entity	1	1	9	9	19
13	Implement Assignment functions	1	1	10	9	20
14	Implement Turn management logic	3	1	11	9	21
15	Implement Priority rules engine	2	1	12	9	22
16	Implement OTP logic for first login	1	1	10	9	20
17	Implement User profile setup/editing	1	1	11	9	21
18	Implement Authentication and Authorization	2	1	12	9	22
19	Turn request UI and QR receipt	3	1	13	9	23
20	QR scan and Turn tracking	2	1	14	9	24
21	Display turns on screen	1	1	15	9	25
22	Time estimation and Web notifications	2	1	15	9	25
23	Permissions management for admins	1	1	13	9	23
24	Admin dashboard (CRUD)	2	1	14	9	24
25	Report generation and export	2	1	15	9	25
26	Error reporting system	1	1	15	9	25
27	Unit testing for core modules	3	1	16	9	26
28	System and integration testing	3	1	17	9	27
29	Deploy to test environment	1	1	18	9	28
30	UAT and environment validation	1	1	19	9	29
31	Deploy to production	1	1	20	9	30

Table 8: Scheduling table with duration, resources and timing constraints

9.2 Time-Limited Scheduling

To better understand the theoretical project duration, a Time-Limited Scheduling (TLS) simulation was first developed. This scenario assumes unlimited resource availability and prioritizes completing tasks as early as possible, based solely on dependencies. The AoA analysis indicated that the project could be completed in 21 weeks under these conditions, revealing a global float of 9 weeks within the planned 30-week timeframe. The TLS schedule was also smoothed to provide a more balanced distribution of workload, despite the idealized nature of the scenario.

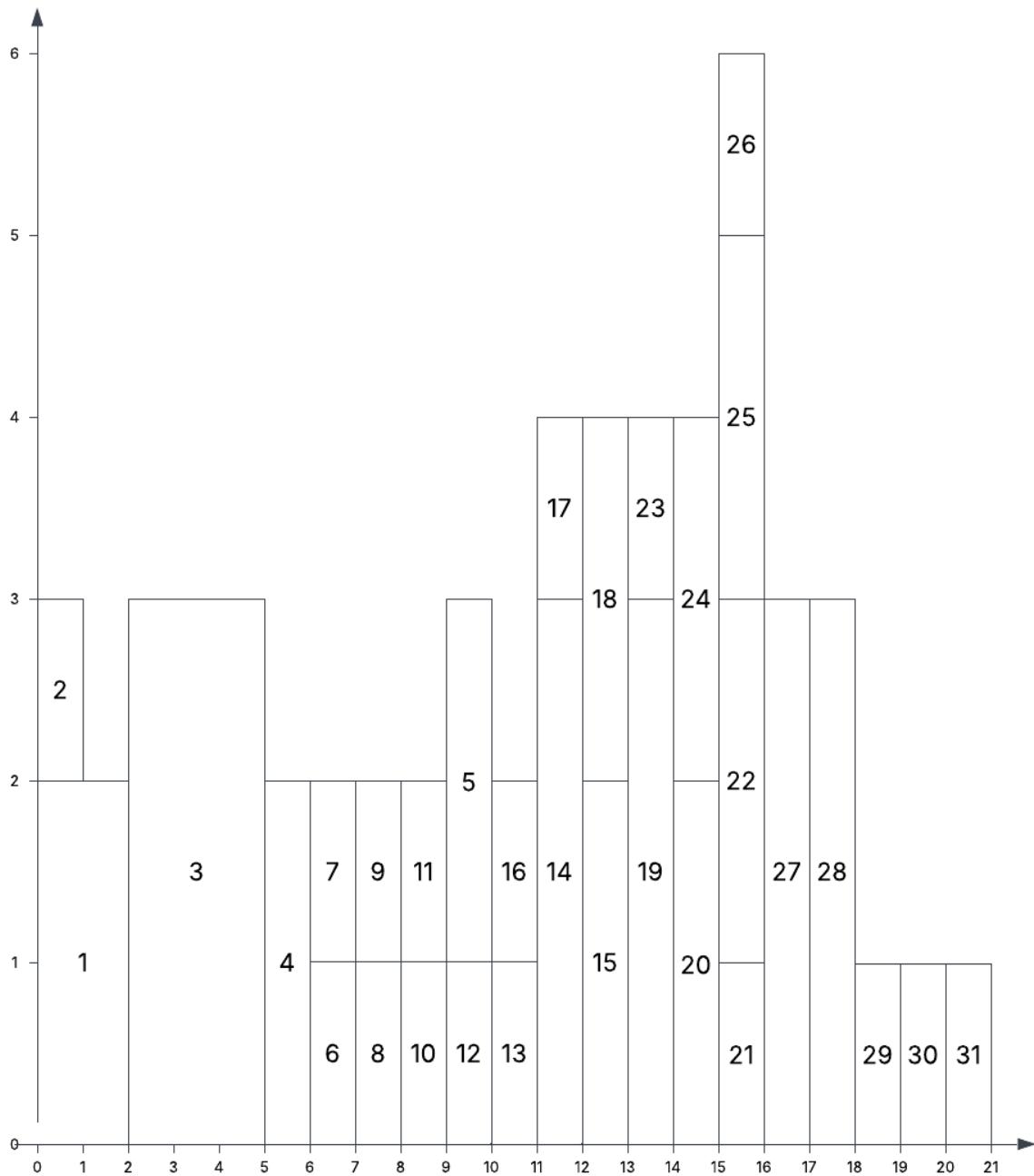


Figure 2: Time-Limited Scheduling (TLS) diagram — smoothed for balanced resource usage.

9.3 Resource-Limited Scheduling

However, the actual scheduling strategy applied in this project is Resource-Limited Scheduling (RLS). Given the course-imposed constraint of working with only three developers and no access to external assistance, the project had to be planned with strict resource availability in mind. The RLS diagram reflects this realistic context, ensuring that tasks are assigned according to team capacity and avoiding overallocation. Like the TLS version, the RLS schedule was also smoothed to distribute workload evenly, aiming to maintain feasibility while respecting the final delivery date of January 9, 2026.

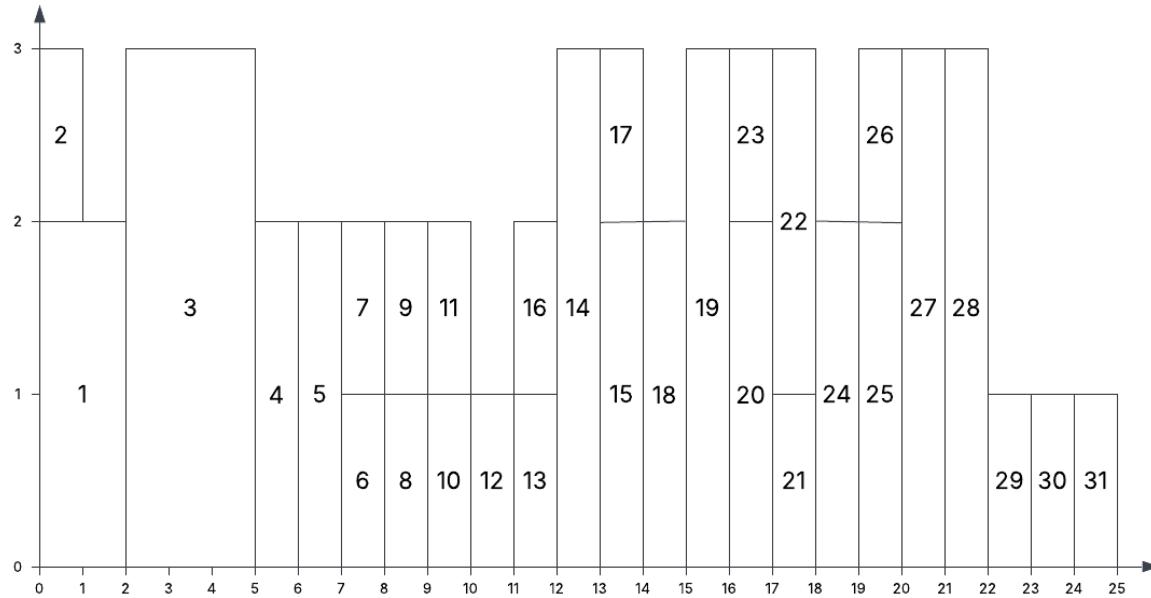


Figure 3: Resource-Limited Scheduling (RLS) diagram — smoothed to reflect realistic resource constraints.

10. Prototype

This section presents the evidence of the interactive prototype developed in Figma for the Queue Management System. Each page includes a brief explanation of the screens showed and its corresponding visual design.

10.1 Page 1: Success

Display: [Figma prototype - Page "Success"]

Description: The touchscreen kiosk begins with a welcome screen that invites users to tap and start the process. After this, the user selects the appropriate service area and proceeds to the turn generation options, where they can choose to create a new turn or retrieve an existing one. Depending on how the system has been configured by the administrator, the user may be required to enter personal information, such as an identification number, or they may be able to generate a turn directly without providing any data. Once the necessary steps are completed, the system generates the turn successfully and displays a confirmation screen with the turn number, estimated wait time, and a QR code for tracking, ensuring users have all the details they need to manage their queue position effectively.

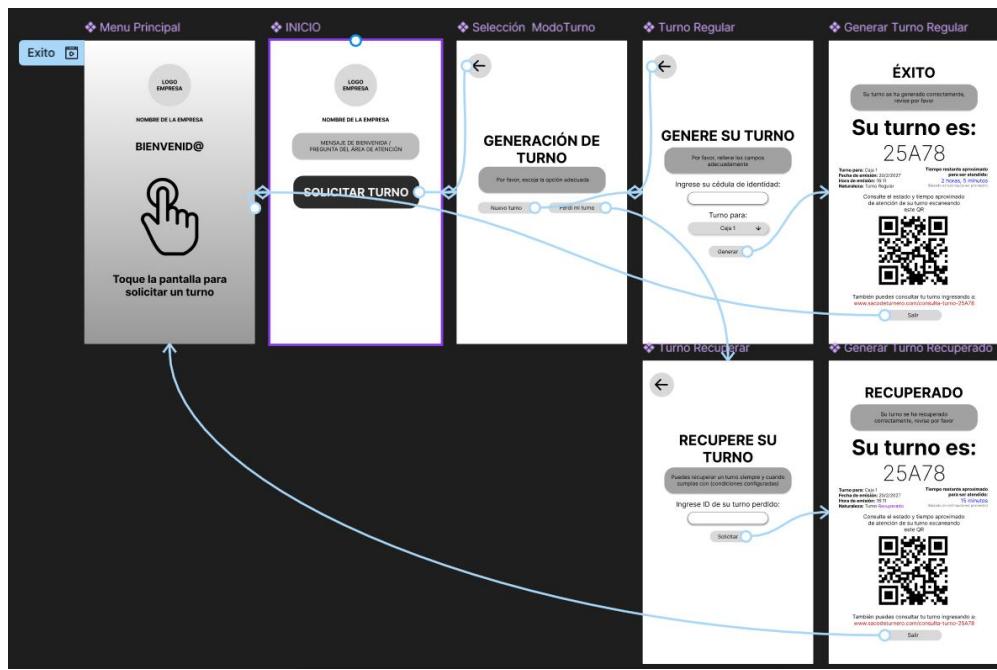


Figure 4: Prototype - Page 1: Success of the touch kiosk.

10.2 Page 2: Failure

Display: [Figma prototype - Page "Failure"]

Description - flow 1: This flow shows what happens when the user leaves required fields empty during the turn generation or recovery process. After attempting to continue with incomplete information, the system displays an error screen notifying the user that there are empty fields and prompting them to try again.

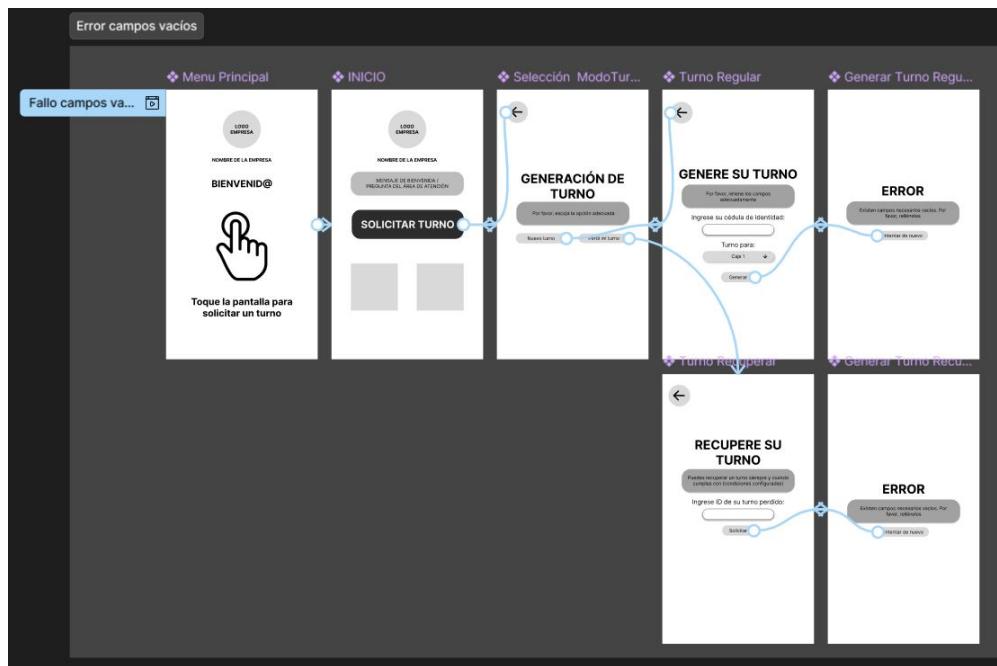


Figure 5: Prototype - Page 2/Flow 1: Empty fields.

Description - flow 2: This flow represents the scenario when the user enters invalid or incorrect information, such as an invalid ID number. When the system detects invalid data, it redirects the user to an error screen that explains the issue and offers an option to re-enter the information correctly.

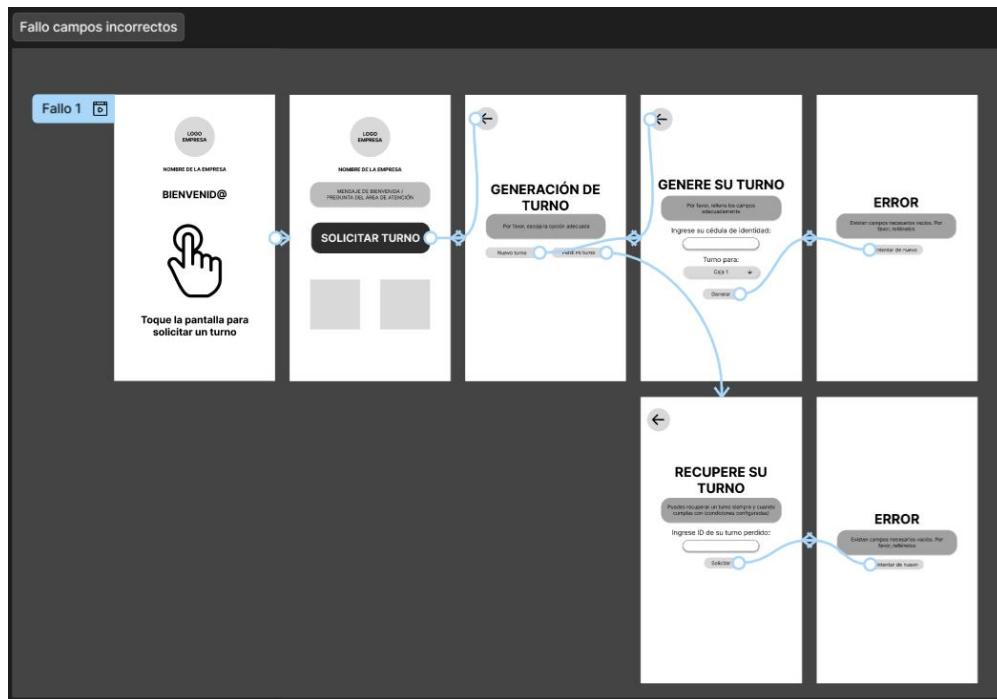


Figure 6: Prototype - Page 2/Flow 2: Invalid information.

10.3 Page 3: TV

Display: [Figma prototype - Page "TV"]

Description: This page shows the TV display used to present the current and upcoming queue numbers to customers, alongside advertising content configured by the administrator. The screens illustrate different scenarios: one where individual turns are called sequentially, another where multiple turns are called simultaneously across different areas or categories depending on business logic, and an example of a simple error case in which the advertising content fails to load correctly. This ensures that customers stay informed about their turn status while the system also displays relevant promotional content.

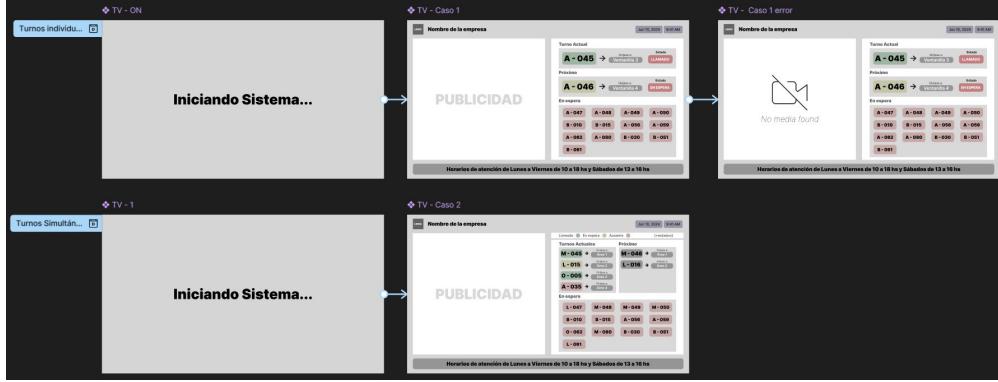


Figure 7: Prototype - Page 3: TV

10.4 Page 4: Administrator

Display: [Figma prototype - Page "Administrator"]

Description: This page shows the administrator's desktop software provided with the queue management system. Through this interface, the administrator can fully manage and customize the system according to their needs: they can create and assign branches, kiosks, and screens; configure which advertising content will be displayed on customer-facing TVs; and adjust the kiosk settings to handle turn generation as desired. The software also allows them to add and manage other administrative users, assign roles and permissions, and oversee all queue operations, ensuring that the entire system is tailored to their specific business logic and customer service flow.

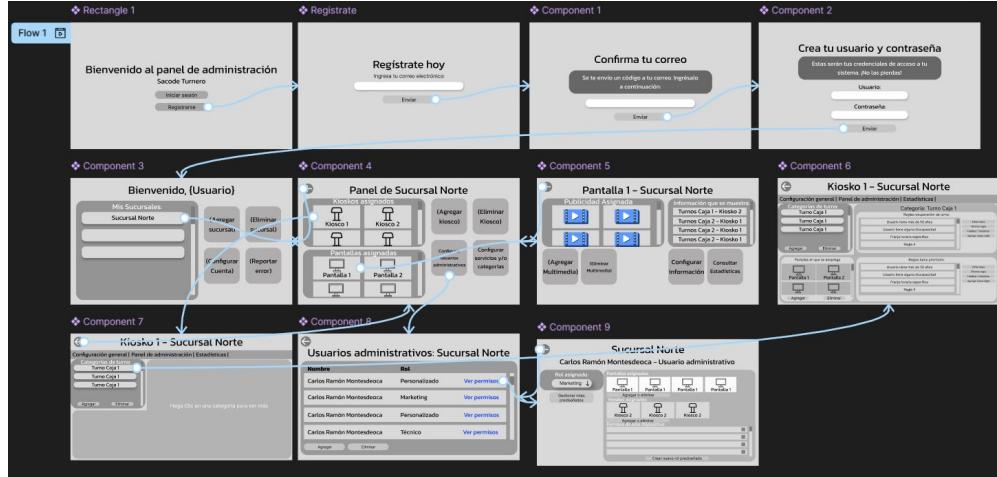


Figure 8: Prototype - Page 4: Administrator

10.5 Page 5: Web User

Display: [Figma prototype - Page "Web User"]

Description: This page shows what a user sees after scanning the QR code printed on their ticket when they take a turn. Through this view, the user can check the real-time status of their queue position at any moment, including dynamic alerts indicating whether their turn is approaching, still has a long wait time, or has already been called but they are late. It also displays an estimated time, the designated area to report to, and a list of helpful reminders to ensure they are prepared when their turn comes up, giving them full control and clear instructions throughout the process.

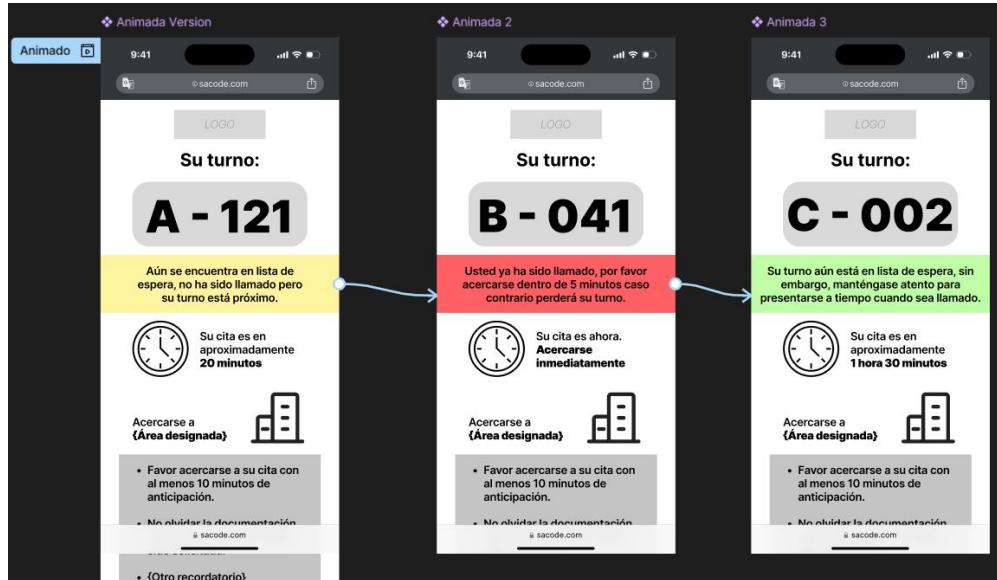


Figure 9: Prototype - Page 5: Web User QR

11. Act of acceptance

This section presents the signed acceptance letter in which the client formally agrees to the proposed solution and the progress achieved so far. This document serves as evidence that the client has reviewed and approved the deliverables at this stage.

ACT OF ACCEPTANCE – SACODE TEAM

Issue Date: 29/06/2025

Project's name: Queue Management System with touch kiosk.

Client's name: Juan Carrillo

Formal acceptance declaration:

The present act serves as evidence that the investigation and development of the sprint zero, with May 21st, 2025, as the starting date and ending June 29th, 2025; and relating to the “QUEUE MANAGEMENT SYSTEM with touch kiosk” project, has been accepted and approved by Ing. Juan Carrillo, who serves as the project’s client; thus, it is concluded that the progress made so far is considered successful considering the client’s needs.

The progress made from the starting date comprises the following points:

- Draft of the software requirements specification, including the organization of functional and non-functional requirements, with validation criteria according to MOSCOW and Sommerville’s classification.
- Initial version of the user stories, documented and reviewed with the client to ensure coverage of core functionalities and expected scenarios.
- Low-fidelity prototype (mockups) representing the main interfaces of the Queue Management System, providing an overview of expected workflows.
- Evidence of requirements elicitation techniques, such as interviews, meetings, and reviews, supporting the specification process.
- Repository with progress, including prototype screens, user stories, and documentation generated during the requirements phase.

Observations:

This acceptance is limited to the progress made in requirements elicitation, user stories development, and the initial prototype. Any further changes or refinements required by the client will be addressed in the next phases. The implementation of the final system will depend on the validated requirements and prototype delivered in this stage.

Client's signature



Scrum master's signature

Accepted by

Ing. Juan Carrillo

Accepted by

Alejandro Sornoza

12. References

References

- [1] SACODE, “sacode-queue-management,” GitHub repository, Available: <https://github.com/vic28code/sacode-queue-management>. [Accessed: 1-Jul-2025].
- [2] IS-Turnero, “SCRUM Project Board,” Jira Software, Available: <https://is-turnero.atlassian.net/jira/software/projects/SCRUM/boards/1>. [Accessed: 1-Jul-2025].
- [3] SACODE, “Queue Management Prototypes,” Figma Project, Available: <https://www.figma.com/design/I3SqsK6MvMDjZkn7mkqLnU/interfaz-kiosko?m=auto&t=IGA3055t7ZE00Lmi-6>. [Accessed: 1-Jul-2025].
- [4] SACODE, ”Evidencias” Google Drive, Available: https://drive.google.com/drive/folders/1cNUGMI2t4dJvHrT84bauvod26ME4W5S0?usp=drive_link. [Accessed: 1-Jul-2025].

13. Annexes

13.1 Annex 1: Excel sheet with user stories

Below is a representative view of the developed user stories. For more details, the complete Excel file (in PDF) with all the stories, their respective scenarios, and acceptance criteria is part of the project's digital deliverable User Stories Queue Management System - SACODE.pdf.

Desarrollo ágil: Historias de usuario y criterios de aceptación

Elaborado por: SACODE

Identificador (ID) de la Historia	Autor	Rol	Enunciado de la Historia			Criterios de Aceptación				Análisis		
			Característica / Funcionalidad	Razón / Resultado	Número (#) de Escenario	Criterio de Aceptación (Título)	Contexto	Evento	Resultado / Comportamiento esperado	Prioridad	Estimación	Sprint
HU-A1	SACODE	Como administrador	Recibir por correo una contraseña temporal (OTP) para ingresar por primera vez	Iniciar el proceso de acceso seguro al sistema	1	OTP enviado correctamente	Administrador ha sido creado por el sistema central	Sistema genera OTP y lo envía al correo asociado	Administrador recibe el OTP y puede usarlo para ingresar por primera vez	MUST	3	1
					2	Fallo en entrega del OTP	El correo del administrador está mal escrito o bloqueado	Sistema intenta enviar el OTP	Sistema registra el error y muestra alerta al usuario central			
HU-A2	SACODE	Como administrador	Usar el OTP recibido para crear la contraseña, completar datos y acceder al sistema	Finalizar el registro del administrador de manera segura	1	El acceso fue exitoso y se configuraron los	El administrador accede por primera vez con un OTP válido	Ingresar OTP, define nueva contraseña, confirma contraseña y	El sistema valida los datos y permite el ingreso al sistema	MUST	5	1
					2	Se bloquea el acceso al insertar	El administrador intenta acceder con	Introduce OTP inválido y presiona 'Confirmar'	El sistema muestra un mensaje de error y bloquea el			
					3	Las contraseñas no coinciden, por tanto el sistema no	El administrador accede con OTP válido pero no coincide la	Llena contraseña y repetir contraseña con valores diferentes	El sistema muestra un mensaje indicando que las contraseñas no coinciden			
					4	Los campos obligatorios están vacíos y por eso no	El administrador accede con OTP válido pero omite campos	Deja vacío campos obligatorios y presiona 'Guardar'	El sistema muestra advertencia de campos requeridos y no guarda			
HU-A3	SACODE	Como administrador	Modificar datos personales y credenciales de acceso	Permitir mantener actualizada la cuenta de administrador	1	Actualización exitosa de datos	Admin accede a su perfil de cuenta	Modifica correo y contraseña actual	Sistema guarda los cambios y confirma la actualización	MUST	3	2
					2	Contraseñas no coinciden	Admin ingresa nueva contraseña y repetición con errores	Presiona 'Guardar cambios'	Sistema muestra validación y evita guardar			
					3	Cambio cancelado por falta de datos	Admin borra el campo de correo y deja vacío el formulario	Presiona 'Guardar cambios'	Sistema muestra advertencia de campos requeridos			
		Como	Crear nuevas	Establecer puntos físicos	1	La sucursal fue creada exitosamente	El administrador accede al módulo de sucursales	Ingresa nombre, dirección, horarios de atención y zona	La sucursal queda registrada y aparece en el listado general			
					2	No se pudo crear la sucursal debido a que faltaron de llenar campos obligatorios	El administrador deja vacío el campo de nombre o dirección	Presiona 'Guardar' sin completar toda la información requerida	El sistema muestra advertencia y no permite continuar			

HU-A4	SACODE	Como administrador	sucursales dentro del sistema	de atención con configuración independiente		3	No se pudo crear la sucursal debido a que se configuró un horario inválido	El administrador coloca una hora de cierre anterior a la de apertura.	Presiona 'Guardar' pero el horario ingresado no es válido	El sistema muestra un mensaje de error de validación	MUST	5	1
						4	No se pudo crear la sucursal debido a que ya existe una duplicada	El administrador intenta crear una sucursal con el mismo nombre o dirección	Presiona 'Guardar' pero ya existía una sucursal idéntica existente	El sistema informa que ya existe una sucursal registrada con esos datos			
HU-A5	SACODE	Como administrador	Crear categorías de atención dentro del sistema	Organizar los turnos según el tipo de servicio requerido		1	Categoría creada exitosamente	Administrador accede al módulo de categorías	Ingrresa nombre, código de letra y color de la categoría	Categoría aparece en el listado disponible para kioscos	MUST	3	1
						2	Faltan campos obligatorios	El nombre o el código están vacíos	Admin presiona 'Guardar'	El sistema muestra advertencia de campos requeridos			
						3	Código duplicado	Se intenta registrar una categoría con el mismo código de letra que otra ya existente	Admin intenta guardar la categoría	El sistema impide la duplicación e informa del conflicto			
HU-A6	SACODE	Como administrador	Crear nuevos kioskos en el sistema	Registrar dispositivos que permitirán la generación de turnos en sucursales		1	El formulario se completa correctamente	El administrador accede al módulo de kioskos con permisos	Ingrresa nombre del kiosko, ubicación, sucursal asociada	El kiosko queda registrado y aparece como disponible en el sistema	MUST	5	1
						2	No se completó el formulario porque faltan campos obligatorios	El administrador intenta crear un kiosko sin completar todos los campos requeridos	Deja vacío uno o más campos obligatorios del formulario y presiona "Guardar"	El sistema no permite que se complete el registro y muestra mensaje de advertencia.			
						3	No se completó el formulario porque nombre duplicado	Ya existe un kiosko con el mismo nombre en la misma sucursal	Admin intenta registrar el nuevo kiosko con nombre repetido	El sistema impide el registro y muestra error de duplicación			
						4	No se completó el formulario porque existe error de conexión a base de datos	El sistema presenta una interrupción en la conexión a la base de datos al guardar el nuevo kiosko	Admin llena el formulario correctamente y presiona "Guardar" pero el sistema pierde conexión.	Se muestra mensaje de error técnico y no se registra el kiosko.			
HU-A7	SACODE	Como administrador	Crear nuevas pantallas para turnos y anuncios institucionales	Mostrar turnos actualizados y anuncios		1	Registro exitoso de una pantalla	El administrador accede al módulo de pantallas	Llena nombre, ubicación y tipo de pantalla y guarda	La pantalla queda registrada y puede asignarse a una sucursal	MUST	5	1
						2	No se completa el registro porque existen campos incompletos	El administrador intenta registrar una pantalla sin seleccionar un tipo de pantalla	Presiona 'Guardar' sin seleccionar un tipo de pantalla	Sistema muestra un mensaje de advertencia indicando que el campo es obligatorio			

			contenido publicitario	instituciones o comerciales	3	No se completa el registro porque hay nombres duplicados en una misma sucursal	Ya existe una pantalla con el mismo nombre en la misma sucursal.	El administrador intenta registrar una nueva pantalla utilizando un nombre duplicado	El sistema rechaza el registro y muestra un mensaje de error indicando duplicidad			
HU-A8	SACODE	Como administrador	Crear contenidos publicitarios para pantallas	Transmitir mensajes informativos o promocionales a los clientes	1	El contenido se sube exitosamente	El administrador accede a módulo de publicidad	Carga archivo de imagen o video y define duración y nombre	El contenido queda disponible para ser asignado a pantallas	MUST	3	2
					2	No hay subida de datos porque existe formato no permitido	El administrador intenta subir un archivo con una extensión no permitida como .exe	Selecciona el archivo no válido y presiona 'Guardar'	El sistema rechaza el archivo y muestra un mensaje de advertencia sobre el formato no permitido			
					3	Se configura erróneamente la duración de la publicidad y por ello no se añade el contenido	El administrador ingresa una duración no válida al intentar registrar el contenido	Intenta guardar el contenido ingresando un valor no numérico como duración (por ejemplo, 'abc')	El sistema muestra un mensaje de error de validación y no permite guardar el contenido			
HU-A9	SACODE	Como administrador	Crear cuentas para usuarios administrativos	Delegar tareas de gestión a otros perfiles autorizados	1	El formulario es válido y se guarda exitosamente	El administrador accede a módulo de usuarios	Llena nombre, correo, rol (técnico, gerente) y sucursal	Cuenta se crea y se asocia a sucursal	MUST	5	2
					2	El correo es inválido, por ello, el formulario no se guarda	El administrador ingresa una dirección de correo sin el carácter '@' o con un formato incorrecto	Presiona 'Crear usuario' con el correo mal ingresado	El sistema muestra un mensaje de error indicando que el correo no es válido			
					3	Rol no seleccionado, por tanto, el formulario no se guarda	El administrador no selecciona un tipo de rol al registrar el usuario	El administrador deja el campo 'Rol' vacío y confirma la creación	El sistema impide guardar y resalta el campo como obligatorio			
					4	Usuario ya existe, entonces el formulario no se guarda	El correo ingresado ya pertenece a otro usuario registrado	El administrador intenta crear una cuenta con un correo duplicado	El sistema muestra un mensaje de error indicando que el correo ya está en uso			
HU-B1	SACODE	Como administrador	Asignar usuarios administrativos a sucursales	Distribuir responsabilidades de gestión local	1	Asignación correcta de usuario	Usuario y sucursal están registrados	El administrador selecciona usuario y sucursal y confirma	Usuario queda asignado y puede acceder a esa sucursal	MUST	4	2
					2	El usuario ya existe en el sistema	El usuario ya se encuentra vinculado a otra sucursal	El administrador intenta asignarlo nuevamente sin desvincularlo	El sistema bloquee la acción y muestra un mensaje indicando conflicto por asignación duplicada			

					3	Falta seleccionar usuario, por ello no se asigna	El administrador no selecciona ningún usuario antes de realizar la acción	Presiona 'Asignar' sin haber elegido un usuario	El sistema muestra un mensaje de advertencia indicando que se debe seleccionar un usuario			
HU-B2	SACODE	Como administrador	Asignar permisos específicos a usuarios administrativos	Controlar el acceso a funcionalidades sensibles dentro del sistema	1	Permiso asignado correctamente	Admin crea un nuevo usuario administrativo	Selecciona permisos como 'modificar pantallas' o 'gestión de kioscos'	El usuario creado solo puede acceder a los módulos autorizados	MUST	4	2
					2	Permiso no guardado por error de validación	Admin no selecciona ningún permiso y presiona guardar	Sistema valida que al menos un permiso debe estar seleccionado	Se muestra advertencia y no se guarda la configuración			
HU-B3	SACODE	Como administrador	Asignar kioscos a sucursales registradas	Permitir que los dispositivos operen en una ubicación específica	1	La asignación de quiosco es exitosa	El kiosko y la sucursal están registrados	Admin selecciona kiosko y sucursal y confirma asignación	El kiosko queda vinculado a la sucursal y habilitado	MUST	3	2
					2	No se asigna el quiosco porque ya fue asignado a otra sucursal	El kiosko ya está asignado a una sucursal activa	El administrador intenta asignarlo a una nueva sucursal sin haber removido la asignación anterior	El sistema bloquea la acción y muestra un mensaje indicando conflicto de asignación			
					3	No se asigna el quiosco porque falta seleccionar sucursal	El administrador solo selecciona el kiosko sin especificar la sucursal	Presiona 'Asignar' sin haber elegido una sucursal	El sistema muestra un mensaje de advertencia indicando que falta información obligatoria			
HU-B4	SACODE	Como administrador	Asignar pantallas a sucursales	Mostrar información de turnos y anuncios en el lugar correcto	1	Asignación exitosa de pantalla y contenido	Hay pantallas disponibles y archivos cargados	Admin selecciona pantalla, sucursal y contenido, y guarda	Pantalla queda asignada y se muestra el contenido	MUST	3	2
					2	Reasignación de pantalla ya vinculada	Pantalla está vinculada a otra sucursal	Admin intenta reasignarla sin liberar	El sistema muestra mensaje de conflicto			
					3	Asignación sin contenido seleccionado	No se selecciona archivo	Admin presiona "Asignar"	Sistema muestra advertencia			
HU-B5	SACODE	Como administrador	Asignar categorías de servicio a kioskos	Ofrecer las opciones correctas según la ubicación	1	Asignación exitosa de categoría	El administrador accede al módulo de configuración del kiosko	Selecciona categoría y la asigna a uno o más kioscos	Categoría queda habilitada en los kioscos asignados	MUST	3	2
					2	La categoría ya está asignada o ya existe en el sistema	La categoría seleccionada ya fue previamente asignada al kiosko	El administrador intenta asignarla nuevamente sin haberla removido	El sistema informa que la categoría ya está asignada y bloquea la acción redundante			

HU-B6	SACODE	Como administrador	Configurar los datos personales requeridos en kioskos	Definir qué información es obligatoria según la sucursal o tipo de atención	1	Configuración con campos personalizados	El administrador accede al módulo de configuración por sucursal	Selecciona campos como cédula, teléfono y nombre como requeridos	Los kioscos de esa sucursal solicitan los datos definidos	COULD	4	3
					2	Desactivación de todos los campos	El administrador desactiva todos los datos personales	Guarda la configuración con todos los campos como 'no requeridos'	Los kioscos no solicitan información adicional al generar turnos			
					3	Error por combinación inválida	El administrador intenta requerir cédula pero desactiva todos los campos visibles	Guarda configuración sin visibilidad del campo obligatorio	Sistema muestra error indicando configuración inconsistente			
HU-B7	SACODE	Como administrador	Configurar reglas de prioridad entre categorías y recuperación de turnos	Definir comportamiento o del sistema en caso de turnos perdidos y orden de atención	1	Configuración de tiempos de recuperación	El administrador accede al módulo de reglas de atención	Establece tiempo máximo (en minutos) para recuperar un turno perdido	Sistema aplica esta regla para todos los kioscos de la sucursal	MUST	5	3
					2	Existe orden de prioridad entre categorías	El administrador configura categorías A, B, C y su jerarquía de atención	Define que A tiene prioridad sobre B y C	Sistema intercala los turnos según la jerarquía configurada			
HU-B8	SACODE	Como administrador	Reinsertar manualmente un turno perdido en la cola	Permitir excepciones ante errores o casos especiales luego del tiempo máximo	1	Se reinserta manualmente	El cliente pide recuperar turno tras pasar tiempo máximo	Administrador lo busca y lo reubica en la cola	Turno vuelve al sistema en la posición indicada por el admin	SHOULD	4	3
					2	El intento de reinserción es inválido	El administrador intenta reinsertar un turno que ya fue marcado como atendido	Selecciona el turno y presiona 'Reincorporar'	Sistema indica que el turno ya fue cerrado y no puede modificarse			
HU-B9	SACODE	Como administrador	Generar y exportar reportes personalizados	Analizar estadísticas de atención y optimizar recursos	1	Generación exitosa de reporte por sucursal	El administrador accede al módulo de reportes y selecciona una sucursal	Define rango de fechas, elige categoría y exporta en PDF	El sistema genera y descarga el archivo correctamente	MUST	5	3
					2	Exportación sin filtros definidos	El administrador accede al módulo pero no define ningún filtro	Presiona 'Generar reporte'	El sistema muestra advertencia solicitando selección de filtros			
					3	Rango de fechas inválido	La fecha de fin es anterior a la fecha de inicio	El administrador selecciona mal el rango y presiona 'Exportar'	El sistema muestra error de validación			

				RECURSOS								
					4	Reporte en formato no soportado	El administrador intenta exportar en formato que no está disponible	Elige 'XLSX' cuando solo está habilitado PDF	El sistema indica que solo está disponible el formato PDF			
					5	Problema de conexión al generar reporte	El servidor de reportes no responde	El administrador intenta generar reporte durante una caída	El sistema informa el error y sugiere intentar más tarde			
HU-B10	SACODE	Como administrador	Reportar errores técnicos o funcionales detectados en el sistema	Facilitar la comunicación de problemas al equipo de soporte	1	Reporte enviado correctamente	Admin experimenta una falla en la interfaz	Accede al formulario de reporte, describe el error y lo envía	El sistema confirma el envío y genera un ticket de seguimiento	SHOULD	3	2
					2	Reporte no enviado por campos vacíos	Admin abre el formulario pero no describe el problema	Presiona 'Enviar' sin completar el campo de descripción	Sistema muestra mensaje de campos requeridos			
HU-C1	SACODE	Como cliente	Generar un turno desde el kiosko	Obtener atención de manera organizada	1	Turno generado exitosamente	El cliente accede a kiosko en horario válido	Selecciona categoría, llena datos requeridos y confirma	Se imprime ticket con QR y número	MUST	5	1
					2	Faltan campos obligatorios	El cliente no llena cédula o nombre (según configuración)	Presiona 'Confirmar' sin completar los campos	El sistema muestra advertencia de campos requeridos			
					3	Kiosko fuera de servicio	El cliente accede a un kiosko sin conexión	Intenta generar un turno	El sistema muestra error técnico y recomienda atención manual			
HU-C2	SACODE	Como cliente que genera un turno	Obtener comprobante impreso del turno con código QR y link de seguimiento	Consultar el estado del turno fácilmente y presentarlo al llegar	1	Impresión automática del turno generado	El cliente genera su turno desde el sistema	El sistema imprime un ticket con código QR y un link único	El cliente recibe el comprobante con toda la información útil del turno	MUST	3	3
					2	Código QR funcional y link accesible	El cliente escanea el código o accede al link desde su móvil	El sistema muestra el estado actualizado del turno (ej. número en cola, tiempo estimado, etc.)	El cliente puede hacer seguimiento desde su dispositivo personal			
					3	Manejo de error en la impresión del ticket	El sistema genera el turno pero falla la impresora	El sistema muestra en pantalla un mensaje con el número de turno, el link de seguimiento, y un botón para reenviar a otra impresora o acudir a recepción	El cliente no pierde el turno y tiene formas alternativas de continuar el proceso			
HU-C3	SACODE	Como cliente	Consultar estado del turno desde	Dar seguimiento sin	1	Se realiza la consulta exitosamente	Cliente escanea QR en navegador con internet	Visualiza su turno, caja y tiempo estimado	Información se presenta actualizada	MUST	3	2

HU-C3	SACODE	Como cliente	dar turno desde código QR	esperar en pantalla fija	2	El QR es inválido	QR fue dañado o mal impreso	Cliente escanea y el sistema no reconoce el código	Sistema muestra mensaje de error	MUST			
HU-C4	SACODE	Como cliente	Visualizar en pantallas cuándo se llama mi turno	Saber con claridad cuándo debo acercarme al punto de atención	1	El turno llamado aparece en pantalla	El cliente se encuentra en la sala y su turno es llamado	La pantalla muestra número, categoría y caja asignada	El cliente identifica que debe acercarse al módulo	MUST	4	3	
					2	El turno no aparece por configuración errónea	La pantalla no está vinculada correctamente a la sucursal	El turno es llamado pero no se refleja en pantalla	El sistema muestra fallback local o indica error técnico				
HU-C5	SACODE	Como cliente	Comprender fácilmente qué categoría elegir en el kiosko	Poder seleccionar la opción correcta sin asistencia	1	Las categorías están claramente descriptas	El cliente está frente al quiosco	Cada botón de categoría incluye íconos o descripción (ej: 'Pagos', 'Consultas')	El cliente elige correctamente sin requerir ayuda	MUST	3	3	
					2	Existe confusión por nombres técnicos o no muy explícitos	Existen categorías con nombres como 'G01' o 'SOP-AUX'	El cliente no sabe cuál seleccionar	El sistema debe permitir configuraciones amigables desde el panel de administración				
HU-C6	SACODE	Como cliente	Visualizar tiempo estimado de atención desde pantallas	Ayudar al cliente a planificar su espera	1	El tiempo es estimado correctamente y es visible para los próximos turnos	Pantalla de la sala muestra la cola	Se calculan promedios por categoría y se muestran	Cliente puede estimar cuánto falta para su turno	SHOULD	3	3	
					2	No hay datos suficientes para estimar correctamente	Recién abierta la sucursal y no hay historial de atención	Pantalla intenta calcular promedio por categoría	Sistema indica que no hay datos suficientes para estimación				
HU-C7	SACODE	Como cliente	Recibir notificaciones web cuando se acerque mi turno	Estar informado sin depender de pantallas	1	Notificación enviada desde la página web	El cliente escanea QR desde navegador compatible (Chrome, Edge)	Permite notificaciones y el sistema le alerta al acercarse su turno	El cliente recibe notificación push con número y caja	COULD	5	3	
					2	iOS no compatible	El cliente usa navegador Safari en iPhone	Intenta activar notificaciones	El sistema informa que su navegador no es compatible				
			Recuperar automáticamente	Dar una segunda	1	Se reincorpora el turno a la cola correctamente	El cliente escanea QR tras perder su turno pero dentro del límite configurado	Presiona 'Reincorporar turno'	El turno vuelve a la cola al final de su categoría				

HU-C8	SACODE	Como cliente	automáticamente e un turno dentro del tiempo permitido	segunda oportunidad en caso de retraso leve	2	No se puede reasignar el turno, ya que está fuera del tiempo máximo permitido para reasignación	El cliente escanea QR después del límite permitido	Presiona 'Reincorporar'	El sistema informa que debe acudir a un administrador	MUST	4	3
HU-C9	SACODE	Como cliente vulnerable	Obtener atención prioritaria sin quedar al principio ni al final	Garantizar accesibilidad sin saltarse completamente la fila	1	Existe un correcto sistema intercalado de selección entre turnos regulares y vulnerables	El sistema detecta un nuevo turno vulnerable generado	El sistema lo inserta de forma balanceada (cada 2 turnos regulares)	Atención prioritaria sin afectar el flujo general	MUST	5	3
					2	Error de asignación de turnos de categoría vulnerable	El quiosco no tiene habilitada categoría vulnerable	Un cliente vulnerable intenta generar un turno y elige una categoría incorrecta	Sistema informa que esa categoría no está disponible y sugiere acudir a recepción			

13.2 Annex 2: User Stories displayed on Jira

This appendix includes a visual snapshot of the list print and the detailed print view of selected User Stories as exported directly from Jira. These views provide structured metadata for each story—such as status, sprint assignment, priority, reporter, and detailed descriptions—allowing traceability and clarity throughout the project. This print format complements the Excel-based documentation by preserving the layout used in the development platform.

Jira

Sorted by: Created descending

1–27 of 27 as at: 30/Jun/25 2:37 AM

T	Summary	Reporter	Status	Priority	Story point estimate
1	Crear categorías de atención dentro del sistema	Alejandro Francisco Sornoza Vera	TO DO	Must	3
2	Asignar permisos específicos a usuarios administrativos	Alejandro Francisco Sornoza Vera	TO DO	Must	5
3	Reportar errores técnicos o funcionales detectados en el sistema	Alejandro Francisco Sornoza Vera	TO DO	Should	2
4	Modificar datos personales y credenciales de acceso	Alejandro Francisco Sornoza Vera	TO DO	Must	3
5	Imprimir comprobante con QR y link de seguimiento	Alejandro Francisco Sornoza Vera	TO DO	Must	3
6	Atención prioritaria para cliente vulnerable	Alejandro Francisco Sornoza Vera	TO DO	Must	5
7	Recuperar turno automáticamente	Alejandro Francisco Sornoza Vera	TO DO	Must	4
8	Notificación web por turno cercano	Alejandro Francisco Sornoza Vera	TO DO	Could	5
9	Ver tiempo estimado de atención	Alejandro Francisco Sornoza Vera	TO DO	Should	3
10	Ver turno llamado en pantalla	Alejandro Francisco Sornoza Vera	TO DO	Must	4
11	Consultar turno por código QR	Alejandro Francisco Sornoza Vera	TO DO	Must	3
12	Generar turno desde kiosko	Alejandro Francisco Sornoza Vera	TO DO	Must	5
13	Exportar reportes personalizados	Alejandro Francisco Sornoza Vera	TO DO	Must	5
14	Reinsertar turno perdido	Alejandro Francisco Sornoza Vera	TO DO	Should	4
15	Configurar reglas de prioridad	Alejandro Francisco Sornoza Vera	TO DO	Must	5
16	Configurar datos requeridos en kiosko	Alejandro Francisco Sornoza Vera	TO DO	Could	4
17	Asignar usuarios administrativos a sucursales	Alejandro Francisco Sornoza Vera	TO DO	Must	4
18	Asignar categorías a kioscos	Alejandro Francisco Sornoza Vera	TO DO	Must	3
19	Asignar pantallas a sucursales	Alejandro Francisco Sornoza Vera	TO DO	Must	3
20	Asignar kioskos a sucursales	Alejandro Francisco Sornoza Vera	TO DO	Must	3

T	Summary	Reporter	Status	Priority	Story point estimate
创	Crear cuentas administrativas	Alejandro Francisco Sornoza Vera	TO DO	Must	5
创	Crear contenido publicitario	Alejandro Francisco Sornoza Vera	TO DO	Must	3
创	Crear pantalla de turnos/publicidad	Alejandro Francisco Sornoza Vera	TO DO	Must	5
创	Crear kiosko	Alejandro Francisco Sornoza Vera	TO DO	Must	5
创	Crear sucursal	Alejandro Francisco Sornoza Vera	TO DO	Must	5
创	Registro de administrador de sucursal	Alejandro Francisco Sornoza Vera	TO DO	Must	5
创	Generar OTP para administrador	Alejandro Francisco Sornoza Vera	TO DO	Must	3

[SCRUM-89] Crear categorías de atención dentro del sistema Created: 30/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0004x:
Sprint:	SCRUM Sprint 1
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero Crear categorías de atención dentro del sistema

[SCRUM-88] Asignar permisos específicos a usuarios administrativos Created: 30/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i00029:
Sprint:	SCRUM Sprint 2
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero asignar permisos específicos a usuarios administrativos

[SCRUM-87] Reportar errores técnicos o funcionales detectados en el sistema Created: 30/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002u:
Sprint:	SCRUM Sprint 4
Story point estimate:	2
Priority:	Should

Description

Como administrador quiero reportar errores técnicos o funcionales detectados en el sistema

[SCRUM-86] Modificar datos personales y credenciales de acceso Created: 30/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002e:
Sprint:	SCRUM Sprint 3
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero modificar datos personales y credenciales de acceso

[SCRUM-85] Imprimir comprobante con QR y link de seguimiento Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002d:v
Sprint:	SCRUM Sprint 3
Story point estimate:	3
Priority:	Must

Description

Como cliente que genera un turno quiero obtener comprobante impreso del turno con código QR y link de seguimiento

[SCRUM-84] Atención prioritaria para cliente vulnerable Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002t:
Sprint:	SCRUM Sprint 4
Story point estimate:	5
Priority:	Must

Description

Como cliente vulnerable quiero obtener atención prioritaria sin quedar al principio ni al final

[SCRUM-83] Recuperar turno automáticamente

Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002q:
Sprint:	SCRUM Sprint 4
Story point estimate:	4
Priority:	Must

Description

Como cliente quiero recuperar automáticamente un turno dentro del tiempo permitido

[SCRUM-82] Notificación web por turno cercano

Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		

Remaining Estimate:	Not Specified
Time Spent:	Not Specified
Original estimate:	Not Specified

Rank:	0 i0002n:
Sprint:	SCRUM Sprint 4
Story point estimate:	5
Priority:	Could

Description

Como cliente quiero recibir notificaciones web cuando se acerque mi turno

[SCRUM-81] Ver tiempo estimado de atención <small>Created: 28/Jun/25 Updated: 30/Jun/25</small>	
Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002m:
Sprint:	SCRUM Sprint 4
Story point estimate:	3
Priority:	Should

Description

Como cliente quiero visualizar tiempo estimado de atención desde pantallas

[SCRUM-79] Ver turno llamado en pantalla <small>Created: 28/Jun/25 Updated: 30/Jun/25</small>	
Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002d:r
Sprint:	SCRUM Sprint 3
Story point estimate:	4
Priority:	Must

Description

Como cliente quiero visualizar en pantallas cuándo se llama mi turno

[SCRUM-78] Consultar turno por código QR Created: 28/Jun/25 Updated: 30/Jun/25			
Status:	To Do		
Project:	Sistema Turnero		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002d:m
Sprint:	SCRUM Sprint 3
Story point estimate:	3
Priority:	Must

Description

Como cliente quiero consultar estado del turno desde código QR

[SCRUM-77] Generar turno desde kiosko Created: 28/Jun/25 Updated: 30/Jun/25			
Status:	To Do		
Project:	Sistema Turnero		

Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002d:i
Sprint:	SCRUM Sprint 3
Story point estimate:	5
Priority:	Must

Description

Como cliente quiero generar un turno desde el kiosko

[SCRUM-76] Exportar reportes personalizados Created: 28/Jun/25 Updated: 30/Jun/25			
Status:	To Do		
Project:	Sistema Turnero		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002l:
Sprint:	SCRUM Sprint 4
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero generar y exportar reportes personalizados

[SCRUM-75] Reinsertar turno perdido Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002 :
Sprint:	SCRUM Sprint 4
Story point estimate:	4
Priority:	Should

Description

Como administrador quiero reinsertar manualmente un turno perdido en la cola

[SCRUM-74] Configurar reglas de prioridad Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i00028 :
Sprint:	SCRUM Sprint 2
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero configurar reglas de prioridad entre categorías y recuperación de turnos

[SCRUM-73] Configurar datos requeridos en kiosko Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002f:
Sprint:	SCRUM Sprint 4
Story point estimate:	4
Priority:	Could

Description

Como administrador quiero configurar los datos personales requeridos en kioskos

[SCRUM-72] Asignar usuarios administrativos a sucursales Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i00027:
Sprint:	SCRUM Sprint 2

Story point estimate:	4
Priority:	Must

Description

Como administrador quiero asignar usuarios administrativos a sucursales

[SCRUM-71] Asignar categorías a kioscos Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0001z:
Sprint:	SCRUM Sprint 2
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero asignar categorías de servicio a kioskos

[SCRUM-70] Asignar pantallas a sucursales Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		

Original estimate:	Not Specified
Rank:	0 i0001r:
Sprint:	SCRUM Sprint 2
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero asignar pantallas a sucursales

[SCRUM-69] Asignar kioskos a sucursales Created: 28/Jun/25 Updated: 30/Jun/25	
Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0001j:
Sprint:	SCRUM Sprint 2
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero asignar kioskos a sucursales registradas

[SCRUM-68] Crear cuentas administrativas Created: 28/Jun/25 Updated: 30/Jun/25	
Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned

Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i00073:
Sprint:	SCRUM Sprint 1
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero crear cuentas para usuarios administrativos

[SCRUM-67] Crear contenido publicitario <small>Created: 28/Jun/25 Updated: 30/Jun/25</small>			
Status:	To Do		
Project:	Sistema Turnero		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i00030:
Sprint:	SCRUM Sprint 1
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero crear contenidos publicitarios para pantallas

[SCRUM-66] Crear pantalla de turnos/publicidad <small>Created: 28/Jun/25 Updated: 30/Jun/25</small>			
Status:	To Do		
Project:	Sistema Turnero		
Components:	None		
Affects versions:	None		

Fix versions:	None
----------------------	------

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0004i:
Sprint:	SCRUM Sprint 1
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero crear nuevas pantallas para turnos y contenido publicitario

[SCRUM-65] Crear kiosko Created: 28/Jun/25 Updated: 30/Jun/25			
Status:	To Do		
Project:	Sistema Turnero		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0005c:
Sprint:	SCRUM Sprint 1
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero crear nuevos kioskos en el sistema

[SCRUM-64] Crear sucursal Created: 28/Jun/25 Updated: 30/Jun/25
--

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i00067:
Sprint:	SCRUM Sprint 1
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero crear nuevas sucursales dentro del sistema

[SCRUM-63] Registro de administrador de sucursal Created: 28/Jun/25 Updated: 30/Jun/25			
Status:	To Do		
Project:	Sistema Turnero		
Components:	None		
Affects versions:	None		
Fix versions:	None		

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 i0002d:
Sprint:	SCRUM Sprint 3
Story point estimate:	5
Priority:	Must

Description

Como administrador quiero usar el OTP recibido para crear la contraseña, completar datos y acceder al sistema

[SCRUM-62] Generar OTP para administrador Created: 28/Jun/25 Updated: 30/Jun/25

Status:	To Do
Project:	Sistema Turnero
Components:	None
Affects versions:	None
Fix versions:	None

Type:	Story	Priority:	Medium
Reporter:	Alejandro Francisco Sornoza Vera	Assignee:	Unassigned
Resolution:	Unresolved	Votes:	0
Labels:	None		
Remaining Estimate:	Not Specified		
Time Spent:	Not Specified		
Original estimate:	Not Specified		

Rank:	0 hzzzzz:
Sprint:	SCRUM Sprint 0
Story point estimate:	3
Priority:	Must

Description

Como administrador quiero recibir por correo una contraseña temporal (OTP) para ingresar por primera vez

Generated at Mon Jun 30 07:37:26 UTC 2025 by Fabricio Chang using Jira 1001.0.0-SNAPSHOT#100285-rev:3d474fc57e349e3dbfa506166be53a4fc5370310.

13.3 Annex 3: Prototype flows from Figma

The corresponding evidence of the proper flows are found in the reference [3] of this document.

Project External Resources

Figma Prototype

Kiosk Interface in case of Success

Click here

The touchscreen kiosk welcomes the user to start, then lets them select a service area and choose to create or retrieve a turn. Depending on the setup, they may enter personal info or generate a turn directly. Finally, the system shows the turn number, wait time, and a QR code for tracking.



LOGO
EMPRESA

NOMBRE DE LA EMPRESA

BIENVENID@



**Toque la pantalla para
solicitar un turno**



LOGO
EMPRESA

NOMBRE DE LA EMPRESA

MENSAJE DE BIENVENIDA /
PREGUNTA DEL ÁREA DE ATENCIÓN

SOLICITAR TURNO



GENERACIÓN DE TURNO

Por favor, escoja la opción adecuada

Nuevo turno

Perdí mi turno



GENERE SU TURNO

Por favor, rellene los campos
adecuadamente

Ingrese su cédula de identidad:



Turno para:



Caja 1



Generar

ÉXITO

Su turno se ha generado correctamente,
revise por favor

Su turno es:
25A78

Turno para: Caja 1
Fecha de emisión: 20/2/2027
Hora de emisión: 16:11
Naturaleza: Turno Regular

Tiempo restante aproximado
para ser atendido:
2 horas, 5 minutos
(Basado en estimaciones promedio)

Consulte el estado y tiempo aproximado
de atención de su turno escaneando
este QR



También puedes consultar tu turno ingresando a:
www.sacodelturnero.com/consulta-turno-25A78

Salir

If a customer loses their turn, they can tap “I lost my turn”, enter the lost turn ID, and press “Request”. If the conditions are met, the system confirms the recovered turn, shows the new turn number, estimated wait time, and a QR code for tracking.



LOGO
EMPRESA

NOMBRE DE LA EMPRESA

BIENVENID@



**Toque la pantalla para
solicitar un turno**



LOGO
EMPRESA

NOMBRE DE LA EMPRESA

MENSAJE DE BIENVENIDA /
PREGUNTA DEL ÁREA DE ATENCIÓN

SOLICITAR TURNO

GENERACIÓN DE TURNO

Por favor, escoja la opción adecuada

Nuevo turno

Perdí mi turno

RECUPERE SU TURNO

Puedes recuperar un turno siempre y cuando cumplas con (condiciones configuradas)

Ingrese ID de su turno perdido:



Solicitar

RECUPERADO

Su turno se ha recuperado
correctamente, revise por favor

Su turno es:

25A78

Turno para: Caja 1
Fecha de emisión: 20/2/2027
Hora de emisión: 16:11
Naturaleza: Turno Recuperado

**Tiempo restante aproximado
para ser atendido:**
15 minutos
(Basado en estimaciones promedio)

Consulte el estado y tiempo aproximado
de atención de su turno escaneando
este QR



También puedes consultar tu turno ingresando a:
www.sacodeturnero.com/consulta-turno-25A78

[Salir](#)

Kiosk Interface in case of Failure

Click here

This flow shows what happens when the user leaves required fields empty during the turn generation or recovery process. After attempting to continue with incomplete information, the system displays an error screen notifying the user that there are empty fields and prompting them to try again.



The flow keeps the same for both cases:



GENERACIÓN DE TURNO

Por favor, escoja la opción adecuada

Nuevo turno

Perdí mi turno



GENERE SU TURNO

Por favor, rellene los campos
adecuadamente

Ingrese su cédula de identidad:

Turno para:

Caja 1 ▾

Generar

GENERACIÓN DE TURNO

Por favor, escoja la opción adecuada

Nuevo turno

Perdí mi turno

RECUPERE SU TURNO

Puedes recuperar un turno siempre y cuando cumplas con (condiciones configuradas)

Ingrese ID de su turno perdido:



Solicitar

This screen would be shown:

ERROR

Existen campos necesarios vacíos. Por favor, rellénelos

Intentar de nuevo

This flow represents the scenario when the user enters invalid or incorrect information, such as an invalid ID number. When the system detects invalid data, it redirects the user to an error screen that explains the issue and offers an option to re-enter the information correctly.



The flow keeps the same for both cases:

GENERACIÓN DE TURNO

Por favor, escoja la opción adecuada

Nuevo turno

Perdí mi turno

GENERE SU TURNO

Por favor, rellene los campos
adecuadamente

Ingrese su cédula de identidad:



Turno para:



Generar



GENERACIÓN DE TURNO

Por favor, escoja la opción adecuada

Nuevo turno

Perdí mi turno



RECUPERE SU TURNO

Puedes recuperar un turno siempre y cuando cumplas con (condiciones configuradas)

Ingrese ID de su turno perdido:



Solicitar

This screen would be shown:

ERROR

La información ingresada es incorrecta.
Por favor, llenar los campos adecuadamente.

Intentar de nuevo

Administrator Pannel

[Click here](#)

This page shows the administrator's desktop software provided with the queue

management system. Through this interface, the administrator can fully manage and customize the system according to their needs: they can create and assign branches, kiosks, and screens; configure which advertising content will be displayed on customer-facing TVs; and adjust the kiosk settings to handle turn generation as desired. The software also allows them to add and manage other administrative users, assign roles and permissions, and oversee all queue operations, ensuring that the entire system is tailored to their specific business logic and customer service flow.



Confirma tu correo

Se te envío un código a tu correo. Ingrésalo
a continuación:

Enviar

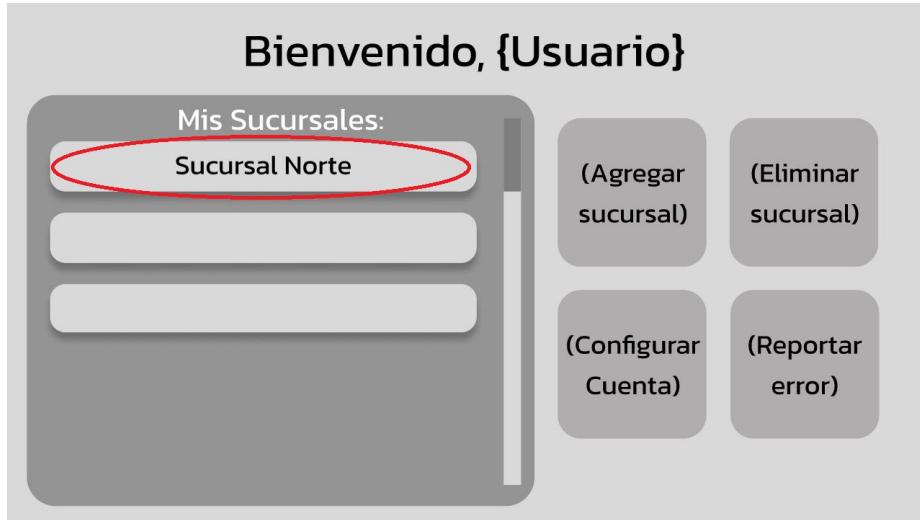
Crea tu usuario y contraseña

Estas serán tus credenciales de acceso a tu
sistema. ¡No las pierdas!

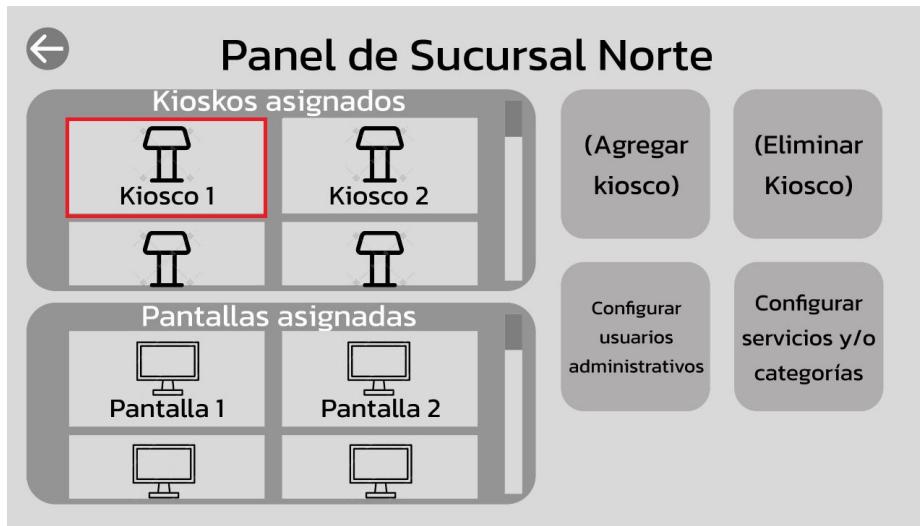
Usuario:

Contraseña:

Enviar



Here we have three different flows shown, this is the first one for kiosks:



Kiosko 1 – Sucursal Norte

Configuración general | Panel de administración | Estadísticas |

Categorías de turno

- Turno Caja 1
- Turno Caja 1
- Turno Caja 1

Agregar Eliminar

Haga Clic en una categoría para ver más

Kiosko 1 – Sucursal Norte

Configuración general | Panel de administración | Estadísticas |

Categorías de turno

- Turno Caja 1
- Turno Caja 1
- Turno Caja 1

Agregar Eliminar

Pantallas en que se despliega

Pantalla 1	Pantalla 2

Agregar Eliminar

Categoría: Turno Caja 1

Reglas recuperación de turno:

- Usuario tiene más de 50 años
- Usuario tiene alguna discapacidad
- Franja horaria específica

Regla 4

Reglas turno prioritario:

- Usuario tiene más de 50 años
- Usuario tiene alguna discapacidad
- Franja horaria específica

Regla 4

Then we return to the branch page to see the next flow for screens:

Panel de Sucursal Norte

Kioskos asignados

Kiosco 1	Kiosco 2

(Agregar kiosco) (Eliminar Kiosco)

Pantallas asignadas

Pantalla 1	Pantalla 2

Configurar usuarios administrativos Configurar servicios y/o categorías

Pantalla 1 - Sucursal Norte

Publicidad Asignada

Play icon	Play icon
Play icon	Play icon

Información que se muestra:

- Turnos Caja 1 – Kiosko 2
- Turnos Caja 2 – Kiosko 1
- Turnos Caja 2 – Kiosko 1
- Turnos Caja 2 – Kiosko 1

(Agregar Multimedia) (Eliminar Multimedia)

Configurar información Consultar Estadísticas

Then we return to the branch page to see the next flow for administrative users:

Panel de Sucursal Norte

Kioskos asignados

Kiosco 1	Kiosco 2

(Agregar kiosco) (Eliminar Kiosco)

Pantallas asignadas

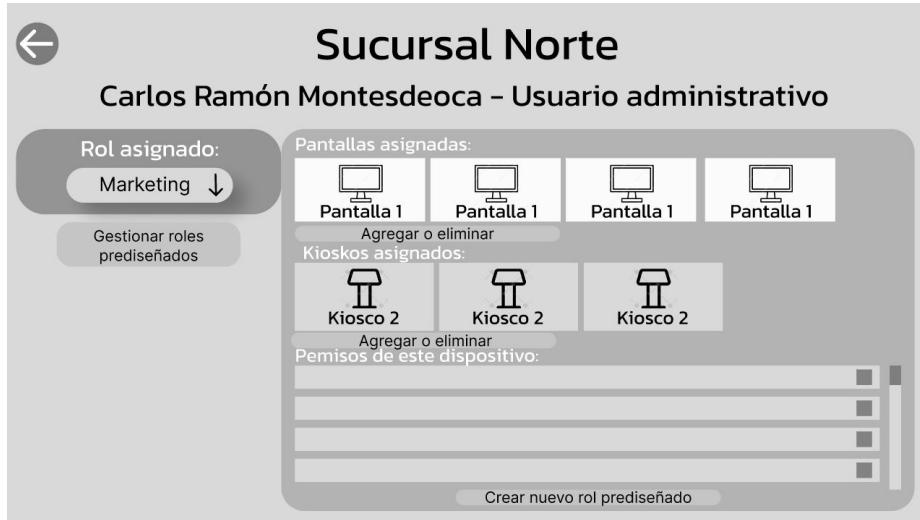
Pantalla 1	Pantalla 2

Configurar usuarios administrativos Configurar servicios y/o categorías

← Usuarios administrativos: Sucursal Norte

Nombre	Role	Ver permisos
Carlos Ramón Montesdeoca	Personalizado	Ver permisos
Carlos Ramón Montesdeoca	Marketing	Ver permisos
Carlos Ramón Montesdeoca	Personalizado	Ver permisos
Carlos Ramón Montesdeoca	Técnico	Ver permisos

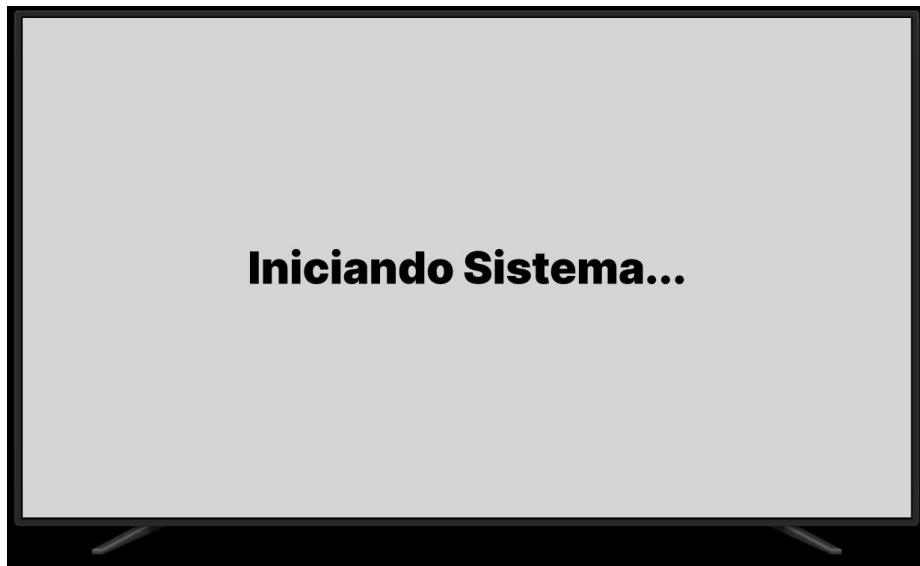
Agregar Eliminar



TV interface

Click here

This page shows the TV display used to present the current and upcoming queue numbers to customers, alongside advertising content configured by the administrator.



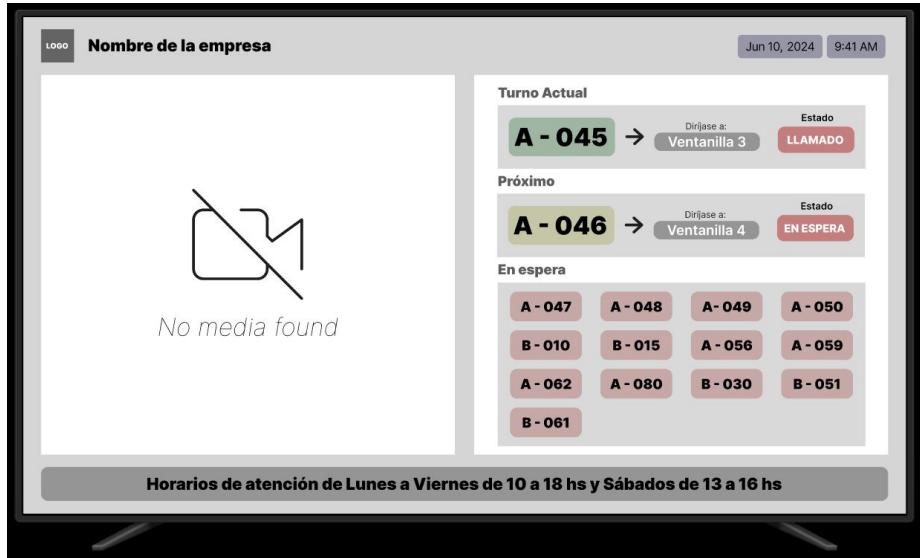
The screens illustrate different scenarios: one where individual turns are called sequentially,



another where multiple turns are called simultaneously across different areas or categories depending on business logic



and an example of a simple error case in which the advertising content fails to load correctly.



User interface QR

Click here

This page shows what a user sees after scanning the QR code printed on their ticket when they take a turn. Through this view, the user can check the real-time status of their queue position at any moment, including dynamic alerts indicating whether their turn is approaching, still has a long wait time, or has already been called but they are late. It also displays an estimated time, the designated area to report to, and a list of helpful reminders to ensure they are prepared when their turn comes up, giving them full control and clear instructions throughout the process.

Approaching turn:



Already called turn:



Long wait time turn:



These were another displays made for a more technical observation of the user web:

The image displays two side-by-side screenshots of a mobile application interface, likely for a service counter or appointment system. Both screens show the same basic information but with different visual styles.

Left Screen (Turn Management):

- Header:** Shows the time (9:41), signal strength, and battery level.
- URL:** sacode.com
- Logo:** A small placeholder icon.
- Text:** **Su turno:** (Your Turn)
- Turn Number:** **A - 121**
- Turn Status:** Estado del turno: En espera de ser llamado. (Yellow dot icon)
- Turns Remaining:** Turnos faltantes: Faltan aproximadamente 4 turnos para que sea atendido.
- Estimated Time:** Tiempo estimado: 20 minutos
- Small Note:** El tiempo estimado es solo un aproximado, no es exacto y puede someterse a cambios.
- Location:** Dirijirse a: Laboratorio 2
- Recommendation:** Recomendación: Llegue con al menos 15 minutos de anticipación para evitar retrasos en su atención y recuerde traer cualquier documento o material que se le haya solicitado.
- Page Bottom:** sacode.com

Right Screen (Turn Management):

- Header:** Shows the time (9:41), signal strength, and battery level.
- URL:** sacode.com
- Turn Number:** **A - 121**
- Turn Status:** Estado del turno: En espera de ser llamado. (Yellow dot icon)
- Turns Remaining:** Turnos faltantes: Faltan aproximadamente 4 turnos para que sea atendido.
- Estimated Time:** Tiempo estimado: 20 minutos
- Small Note:** El tiempo estimado es solo un aproximado, no es exacto y puede someterse a cambios.
- Location:** Dirijirse a: Laboratorio 2
- Recommendation:** Recomendación: Llegue con al menos 15 minutos de anticipación para evitar retrasos en su atención y recuerde traer cualquier documento o material que se le haya solicitado.
- Page Bottom:** sacode.com