

Sistemas con Microprocesadores y Microcontroladores



UNIVERSIDAD NACIONAL DE TUCUMÁN
facet
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍA

“Sistema de gestión para transporte”

Integrantes Grupo 1:

Jimenez, Facundo Ariel

Frías, Lucas Benjamín

Rodriguez, Ulises Agustin

Tutor:

Mg. Ing. Volentini, Esteban

ÍNDICE

1. Introducción.....	3
2. Objetivos.....	3
2.1 Motivaciones y estado del arte.....	3
2.2 Puntos de partida y consideraciones.....	4
2.3 Interfaz.....	5
2.3.1 Interfaz del chofer.....	5
2.3.2 Interfaz del pasajero.....	5
2.4 Funcionamiento.....	6
Pantalla LCD 20x4 (interfaz de pasajero).....	7
Display 7 segmentos interno (interfaz de chofer).....	7
Display de 7 segmentos externo (panel frontal del colectivo).....	7
3. Diseño de Hardware.....	8
3.1 Diseño general.....	8
3.2 Conexionado de los dispositivos.....	10
3.2.1 Lector RFID.....	10
3.2.2 Display LCD.....	11
3.2.3 7 Segmentos Chofer.....	12
3.2.4 7 Segmentos Exterior.....	13
3.2.5 Botones Chofer.....	14
3.2.6 Botones Destino.....	14
3.2.7 LEDs.....	14
3.2.8 GPS.....	15
3.2.9 Memorias.....	16
3.2.9.1 Memoria Flash.....	16
3.2.9.2 Memoria SRAM.....	17
4. Diseño de Software.....	18
4.1 Funcionamiento General.....	18
4.1.1 Diagrama de funcionamiento general.....	18
4.1.2 Tabla de funcionamiento.....	20
4.2 Estructura de datos.....	21
4.2.1 Tabla de estructura de datos.....	21
4.3 Diagramas de flujos de las rutinas.....	23
4.3.1 Diagrama General de Software.....	23
4.3.2 Diagrama General de Interrupciones del Timer.....	24
4.3.3 Diagrama Rutina de Inicialización.....	25
4.3.4 Diagrama Rutina de Actualizar Pantalla.....	26
4.3.5 Diagrama Rutina de Actualizar Ubicación desde GPS.....	27
4.3.6 Diagrama Rutina de Leer Tarjeta.....	28
4.3.7 Diagrama Rutina de Calcular Cobro/Reintegro.....	29
4.3.8 Diagrama Rutina de Seleccionar Destino.....	30
4.3.9 Diagramas Rutinas de Crear y Eliminar Pasajero.....	30
4.3.10 Diagrama Rutina de Leer Botones del Panel del Chofer.....	31
4.3.11 Diagrama Rutina de Reiniciar Viaje.....	32
3. Conclusiones.....	32
5.1 Análisis de costos.....	33
5.2 Mejoras futuras.....	34
5.3 Análisis del uso de DMA.....	34
4. Bibliografía.....	34

1. Introducción

Hace tiempo que el transporte T.E.S.A. de media distancia en Tucumán presenta problemáticas derivadas del control manual que realizan los choferes a la hora de llevar un control del ascenso de pasajeros, la disponibilidad de asientos y la gestión de los destinos. A esto se suma la ineficiencia del cobro en efectivo y la falta de información para determinar cuándo no habrá descensos en determinados poblados. Esta metodología además de generar demoras y errores frecuentes, implica una carga operativa al chofer, el cual debe administrar el cobro, el conteo y organización del recorrido. Al mismo tiempo los pasajeros carecen de información como si la unidad cuenta con asientos disponibles. Todo esto afecta a la calidad general del servicio y la seguridad de los pasajeros.



Para enfrentar las problemáticas mencionadas, proponemos el desarrollo de un sistema que automatice el conteo de pasajeros, gestione eficientemente los destinos y muestre la disponibilidad de asientos libres en tiempo real. Para ello este sistema integra un método de pago electrónico, botones para la selección del destino, servicio de localización por GPS, leds y displays de 7 segmentos, formando un sistema completo e intuitivo.

2. Objetivos

Diseñar un sistema de monitoreo de pasajeros y gestión de destinos en el transporte T.E.S.A.

Dicho dispositivo debe poder:

- Diferenciar el destino de cada pasajero.
- Mostrar cantidad de lugares disponibles si es que los hay, sino indicar que está lleno.
- Determinar si el pasajero está subiendo o bajando de una unidad.

2.1 Motivaciones y estado del arte

En ciudades grandes el uso de validadores y ticketing electrónico es amplio; sin embargo, en transporte interurbano regional la operación manual sigue siendo lo más común: los choferes cobran en efectivo y registran destinos de forma manual en la mayoría de las empresas locales (ej. TESA). Esto crea la ventana de oportunidad para soluciones escalables y de bajo costo que se adapten a la operación interurbana.

En este análisis abordaremos las siguientes consignas:

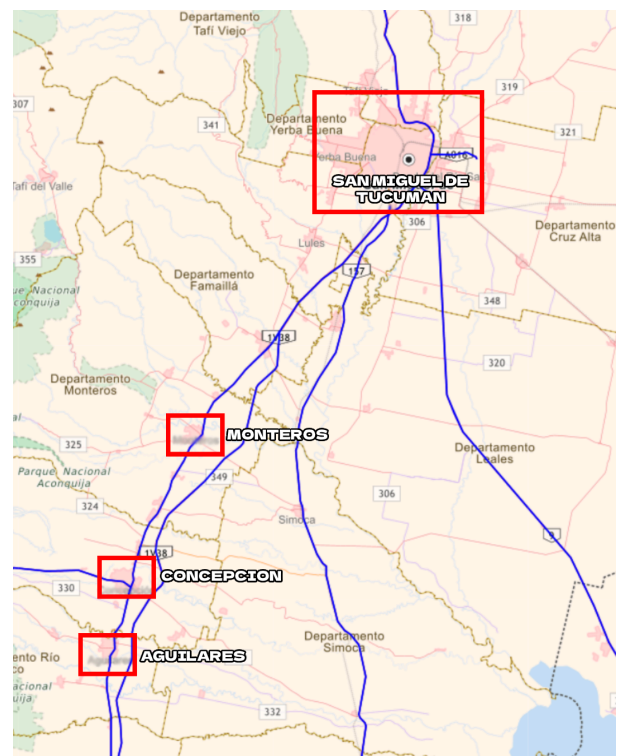
Costo e infraestructura: Las soluciones urbanas “completas” implican inversión y complejidad; hay demanda de soluciones más económicas y específicas para interurbano que no requieran la misma escala de infraestructura.

Adaptación al contexto interurbano: Reglas de cobro (como la tarifa por tramo y los reintegros por descenso anticipado) y la ausencia de pasajeros de pie hacen que la lógica difiera del transporte urbano. Un sistema que implemente cobro anticipado y reintegro automático está alineado con esa operativa.

Valor operativo: Automatizar el conteo y ofrecer al chofer información en tiempo real permite decidir evitar paradas sin pasajeros, optimizar tiempos y costes (combustible, horas hombre) y mejorar la experiencia del pasajero.

2.2 Puntos de partida y consideraciones

- No pueden viajar de pie (el máximo de pasajeros es 44).
- Las tarifas son establecidas por tramos entre ciudades.
- Solo hay paradas dentro de las ciudades.
- El cobro será gestionado por tarjeta mediante RFID.
- La tarifa será máxima en el ingreso y se reintegrará lo correspondiente en el egreso.
- El control de pasajeros se llevará a cabo por software según la detección de tarjeta.
- El chofer contará con un sistema de ajuste manual para casos extraordinarios.
- Se comprende como trayecto:
 - Inicio en S.M.T.
 - Paso por Monteros.
 - Paso por Concepción.
 - Paso por Aguilares (a partir de aquí cambia el sentido del trayecto).
 - Paso por Concepción.
 - Paso por Monteros.
 - Fin en S.M.T.



Nota: El trayecto se llevará a cabo por la Ruta 38 y en base a la información del sistema, el chofer podrá tomar decisiones sobre el recorrido.

2.3 Interfaz

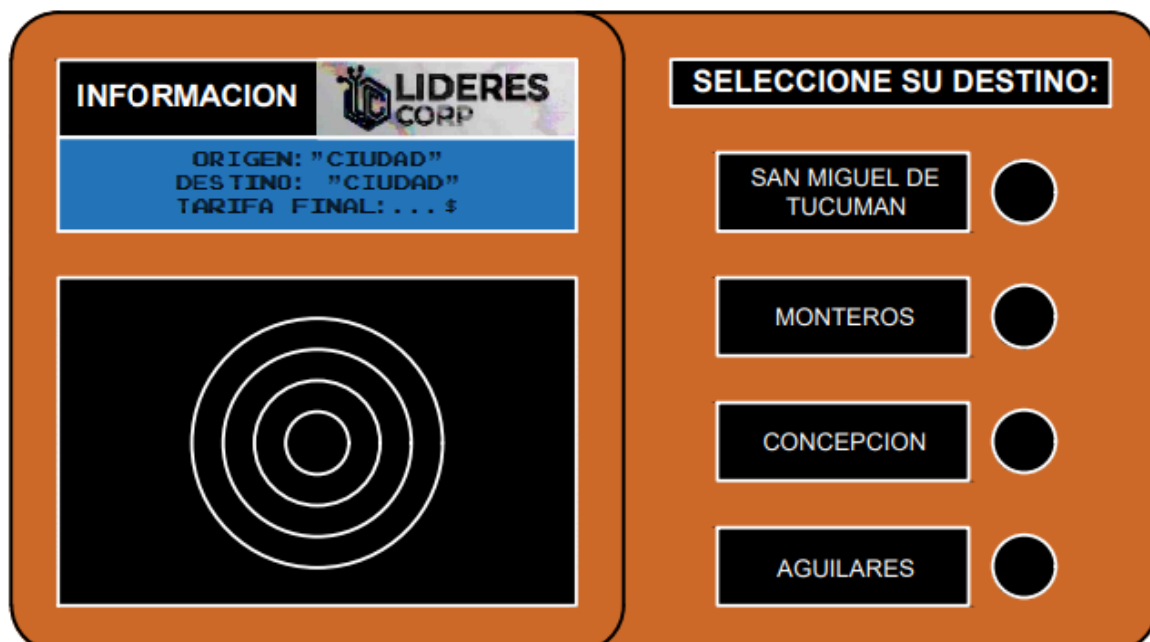
2.3.1 Interfaz del chofer



La interfaz del chofer está compuesta por:

- Botones: 6 botones F1-F4, ACEPTAR y CANCELAR.
- Pantalla: 8 Displays de 7 segmentos.

2.3.2 Interfaz del pasajero



La interfaz del pasajero está compuesta por el módulo de pago y reintegro que consta de:

- Botones: 4 botones para elegir el destino (SMT, Monteros, Concepción o Aguilares).
- Pantalla: Display LCD 20x4
- Lector de tarjeta: Lector RFID MFRC522

INFORMACION



El frente de las unidades estarán conformadas por:

- Indicador de asientos disponibles: 2 Displays de 7 segmentos.
- Indicador de próximo destino: Hay 4 LEDs.

2.4 Funcionamiento

Detección del pasajero y lectura de tarjeta

Cuando un pasajero ingresa a la unidad, acerca su tarjeta al lector RFID en la interfaz de usuario. El lector obtiene el ID único de la tarjeta y lo envía al microcontrolador mediante el bus SPI. El sistema distingue el estado del pasajero en cada interacción si:

- Es la primera lectura de esa tarjeta en el viaje, es decir, el pasajero está ascendiendo.
- La tarjeta ya se encontraba registrada en memoria, es decir, el pasajero está descendiendo.

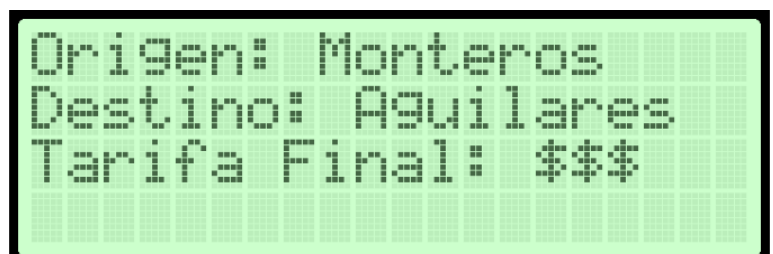
Selección del destino

Si la lectura corresponde a un ascenso, el pasajero selecciona su destino mediante un panel de cuatro botones. Cada botón representa una ciudad del recorrido (S.M.T., Monteros, Concepción, Aguilares). Se guardará temporalmente en memoria junto al ID de la tarjeta, la ciudad de origen (obtenida por el GPS) y ciudad de destino seleccionada (como información para el chofer).

Cobro inicial y actualización del sistema

En el momento del ascenso:

1. El sistema descuenta el valor del pasaje más caro.
2. Actualiza el número de asientos ocupados.
3. Actualiza el contador de pasajeros por destino.
4. Muestra por el Display LCD los datos relevantes:



Al mismo tiempo, los displays de 7 segmentos internos del chofer reflejan la cantidad de personas que van a cada destino. Además, el display grande ubicado en el frente de la unidad muestra la cantidad de asientos libres para que los pasajeros de las paradas puedan verlo.

Nota: Se verifica que el saldo de la tarjeta sea válido (no negativo) antes de realizar el cobro. En caso de no ser válido, no se le permitirá el ingreso al cliente..

Determinación del próximo destino (GPS)

El sistema utiliza un módulo GPS para conocer en todo momento la posición de la unidad. Mediante la comparación de coordenadas:

- Determina en qué ciudad se encuentra la unidad.
- Determina el sentido del viaje.
- Define el próximo destino del recorrido.
- Actualiza cuál LED indicador se prende al frente de la unidad para que los pasajeros puedan saber hacia dónde se dirige.

Descenso del pasajero y reintegro

Al descender, el pasajero vuelve a apoyar su tarjeta en el lector RFID. El sistema reconoce que ya estaba registrada y ejecuta el proceso de descenso:

1. Verifica la ubicación en la que ocurre el descenso.
2. Calcula la tarifa con la diferencia entre las ciudades de ascenso y descenso mediante la tabla de precios.
3. Realiza el reintegro automático del saldo correspondiente en la tarjeta.
4. Actualiza los contadores de pasajeros hacia cada ciudad y asientos disponibles.

Controles del chofer — Ajustes manuales

Por seguridad y tolerancia a fallos, el sistema integra un panel para el chofer con seis botones: **F1, F2, F3, F4, Aceptar y Cancelar**. Con ellos podrá interactuar con un menú que le permitirá reiniciar el sistema, ajustar manualmente el número de personas que se dirigen a cada destino o cambiar manualmente el próximo destino. Esto permite corregir situaciones extraordinarias como fallas momentáneas del lector, pasajeros sin tarjeta o emergencias operativas.

Pantallas

El sistema presenta información útil al chofer y a los pasajeros con distintos indicadores:

Pantalla LCD 20x4 (interfaz de pasajero)

- Origen detectado por GPS.
- Destino seleccionado.
- Tarifa final.

Display 7 segmentos interno (interfaz de chofer)

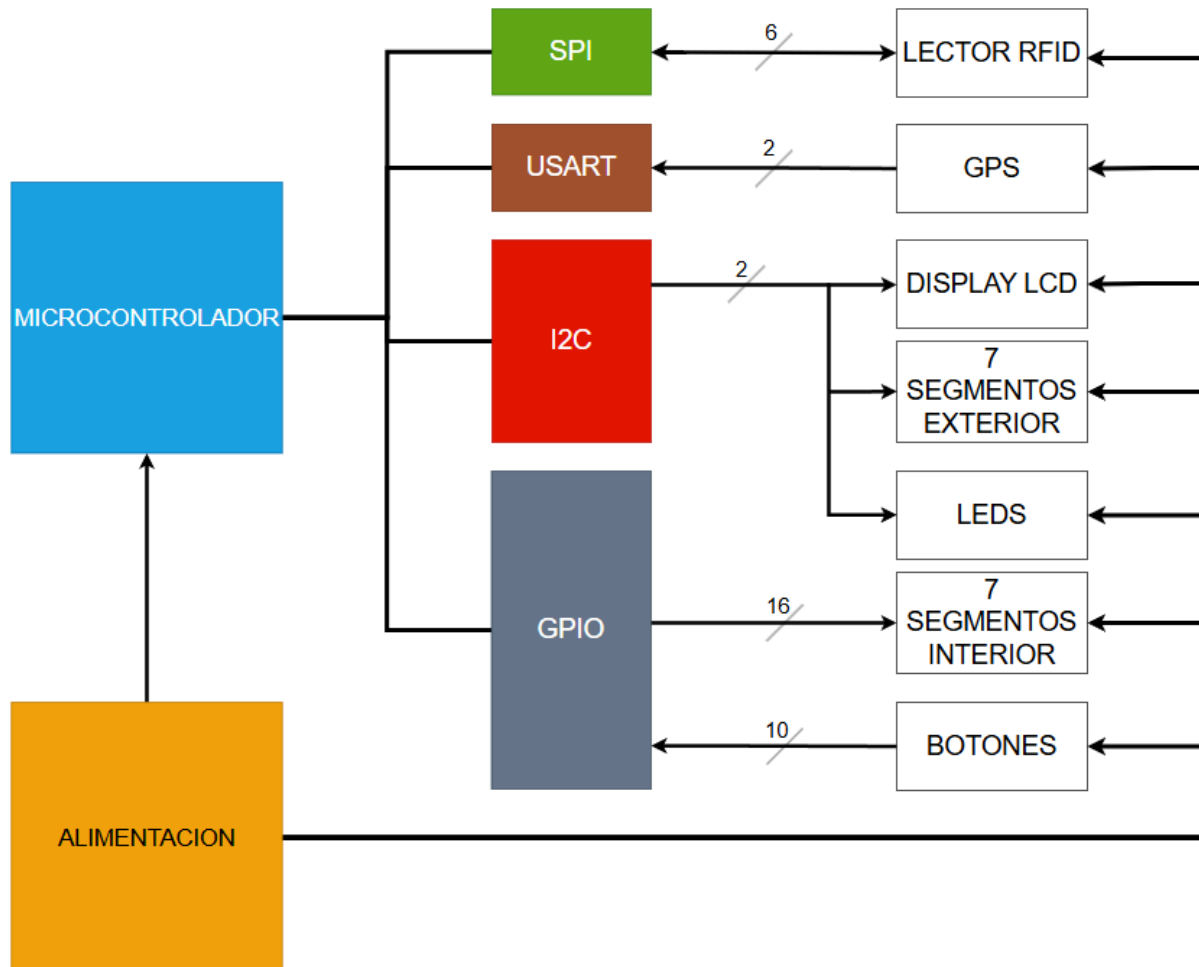
- Cantidad de pasajeros por ciudad.
- Próximo destino.
- Indicador de que la unidad está llena.

Display de 7 segmentos externo (panel frontal del colectivo)

- Cantidad de asientos disponibles para que los pasajeros externos vean si hay lugar.

3. Diseño de Hardware

3.1 Diseño general



Del diseño general podemos destacar que necesitamos 6 líneas SPI, 2 líneas I2C, 2 líneas USART y 26 líneas GPIO.

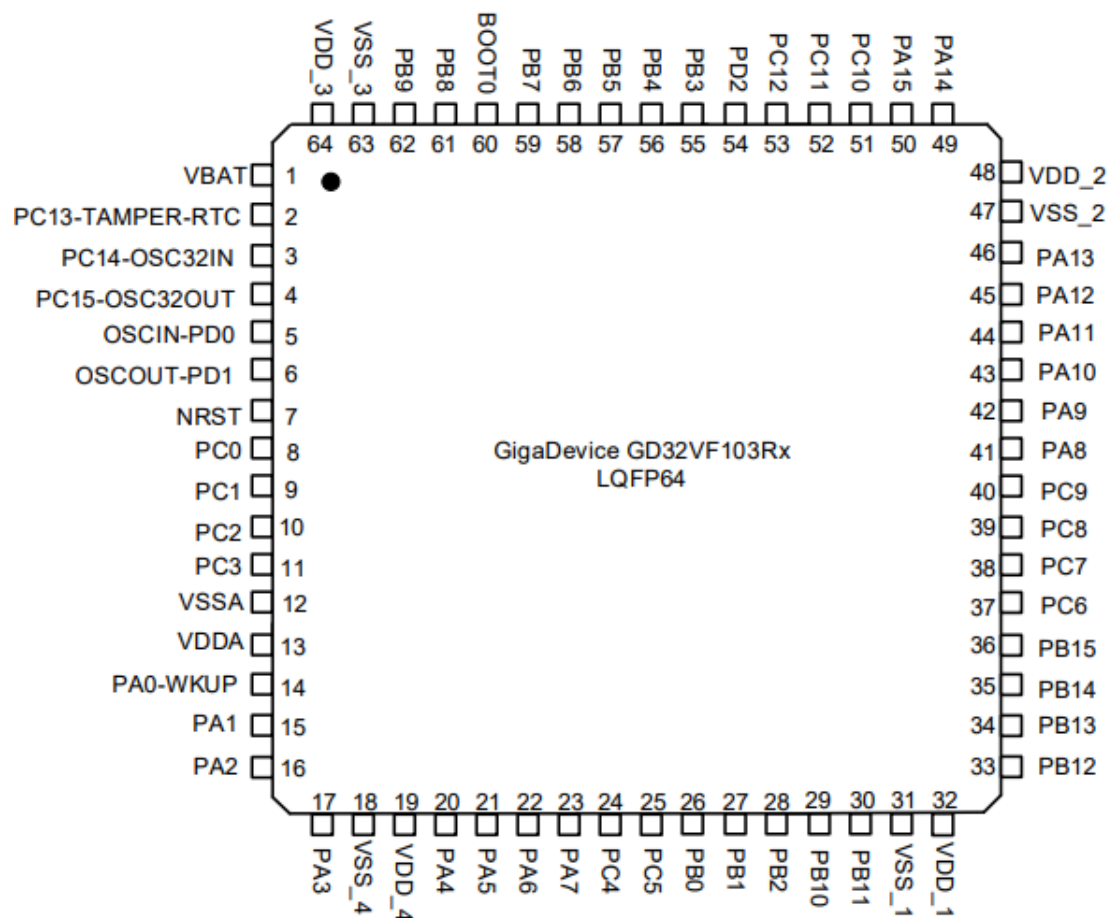
Esta información es circunstancial a la hora de elegir un microprocesador para trabajar, sumado a un análisis de la memoria RAM y flash necesaria.

Considerando que utilizamos un micro **GD32VF103RBT6 LQFP64** que soporta hasta 51 GPIOs, 2 interfaces I2C separadas, 5 USARTs y 3 SPIs, nos encontramos en condiciones de utilizar dicho micro en nuestro proyecto.

Part Number		GD32VF103					
		RB	R8	R6	R4	VB	V8
Flash	Code area (KB)	128	64	32	16	128	64
	Data area (KB)	0	0	0	0	0	0
	Total (KB)	128	64	32	16	128	64
SRAM (KB)		32	20	10	6	32	20
Timers	General timer(16-bit)	4 (1-4)	4 (1-4)	2 (1-2)	2 (1-2)	4 (1-4)	4 (1-4)
	Advanced timer(16-bit)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)	1 (0)
	SysTick	1	1	1	1	1	1
	Basic timer(16-bit)	2 (5-6)	2 (5-6)	2 (5-6)	2 (5-6)	2 (5-6)	2 (5-6)
	Watchdog	2	2	2	2	2	2
	RTC	1	1	1	1	1	1
Connectivity	U(S)ART	5	5	2	2	5	5
	I2C	2 (0-1)	2 (0-1)	1	1	2 (0-1)	2 (0-1)
	SPI/I2S	3/2 (0-2) / (1-2)	3/2 (0-2) / (1-2)	1/-	1/-	3/2 (0-2) / (1-2)	3/2 (0-2) / (1-2)
	CAN	2	2	2	2	2	2
	USBFS	1	1	1	1	1	1
GPIO		51	51	51	51	80	80
EXMC		-	-	-	-	1	1
EXTI		16	16	16	16	16	16
ADC	Units	2	2	2	2	2	2
	Channels	16	16	16	16	16	16
DAC		2	2	2	2	2	2
Package		LQFP64				LQFP100	

3.2 Conexionado de los dispositivos

Partimos del esquema de los puertos que tenemos en nuestro procesador:



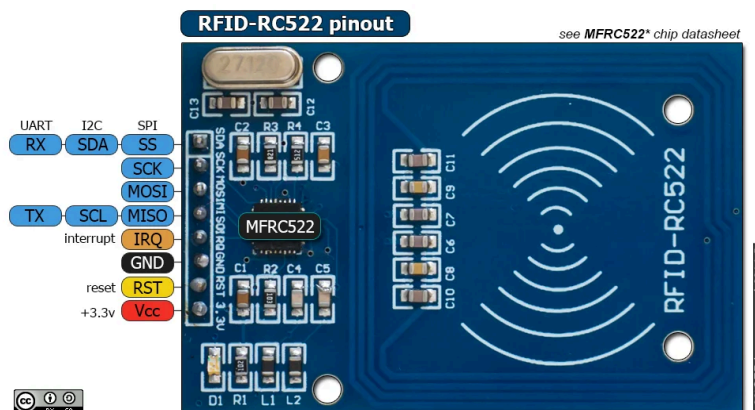
3.2.1 Lector RFID

Seleccionamos el módulo lector/escritor basado en el chip **NXP MFRC522** (MIFARE Classic). Este dispositivo opera mediante **acoplamiento inductivo** en la banda de Alta Frecuencia **13.56 MHz**, cumpliendo con el estándar internacional **ISO/IEC 14443A**.

La elección e integración del hardware se fundamenta con los siguientes aspectos:

- **Compatibilidad Industrial:** Es nativamente compatible con tarjetas de la familia **MIFARE Classic**, tecnología estándar actualmente utilizadas en sistemas de transporte masivo (como tarjeta SUBE o Ciudadana) y control de acceso, lo que garantiza la interoperabilidad del prototipo con medios de pago existentes.
- **Interfaz de Comunicación de Alta Velocidad (SPI):** Se optó por la interfaz **SPI** en lugar de UART o I2C. SPI permite una comunicación sincrónica *Full-Duplex* (bidireccional simultánea) a velocidades de hasta **10 Mbit/s**. Esta velocidad es crítica para realizar la negociación, anticolisión y lectura de la tarjeta en tiempos imperceptibles para el usuario (baja latencia), asegurando una experiencia de uso fluida.
- **Estrategia de Lectura (UID):** El funcionamiento del sistema se basa en la extracción del **UID (Identificador Único)** de 4 bytes de la tarjeta. El hardware realiza

un ciclo de "polling" (búsqueda constante) y, al detectar un campo magnético perturbado por una tarjeta, interroga su número de serie. Este dato se transmite al microcontrolador para ser procesado como una "llave de acceso", evitando la escritura en la memoria interna de la tarjeta.

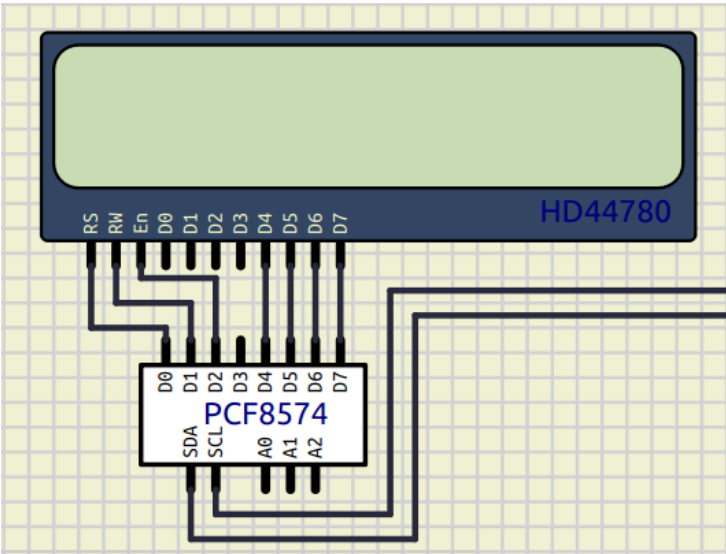


Conexiones de pines:

Pin Módulo	Funcion	Indispensable	Conexión
VCC	+3.3V	Si	Alimentación
GND	Tierra	Si	Tierra
SDA	Chip Select	Si	PA4
SCK	SPI Clock	Si	PA5
MOSI	Master Out Slave In	Si	PA7
MISO	Master In Slave Out	Si	PA6
RST	Reset	Opcional	-
IRQ	Interrupciones	Opcional	-

3.2.2 Display LCD

Elegimos un display LCD 20x4 por la cantidad de líneas de texto que necesitamos mostrar. Internamente trabaja con un controlador HD44780 en modo paralelo de 4 bits, para su conexión utilizamos un expansor I2C basado en PCF8574, para reducir la cantidad de pines utilizados.



Este expansor I2C se conectara de la siguiente manera:

Pin Módulo	Funcion	Conexión
SDA	Datos	PB7 (I2C0_SDA)
SCL	Serial Clock	PB6 (I2C0_SCL)

Luego este se conecta al LCD.

El micro no manda directamente texto ASCII, envía instrucciones HD44780 en formato de 4 bits, por lo que para mandar el Byte que permite I2C debe mandarlo en 2 partes, parte alta y baja, luego en el expansor lo procesa y envía al LCD un pulso habilitador y la información. Para todo esto en el momento de inicializar el display LCD hay que enviar una serie de datos a direcciones específicas del mismo para configurarlo.

3.2.3 7 Segmentos Chofer

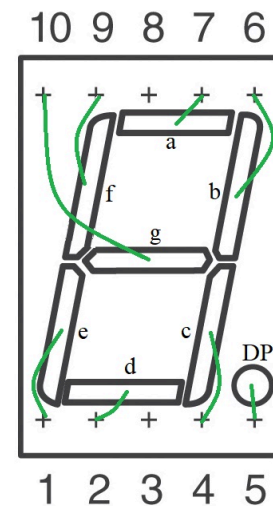
Se seleccionaron displays LED de 7 segmentos SC56-11GWA KB 12-21 M3 que son de ánodo común, ya que ofrecen alta visibilidad y bajo costo. Sin embargo, estos módulos requieren corrientes de entre 15 y 30 mA por segmento y funcionan a 5 V, valores que superan la capacidad de los pines GPIO del micro (3.3 V y pocos miliamperes).

Debido a esto se agrega una etapa de transistores para manejar potencia:

- **Transistores NPN** para controlar los cátodos de cada segmento sin sobrecargar los GPIO.
- **Transistores PNP** para conmutar el ánodo común de cada dígito, ya que deben manejar 5 V y mayor corriente.

Nota: Los dígitos se manejan mediante multiplexado, para reducir consumo y ahorrar pines.

Funcion	Conexión
Segmento A	PC0
Segmento B	PC1
Segmento C	PC2
Segmento D	PC3
Segmento E	PC4
Segmento F	PC5
Segmento G	PC6
Segmento DP	PC7
Dígito 1 ánodo	PD0
Dígito 2 ánodo	PD1
Dígito 3 ánodo	PD2
Dígito 4 ánodo	PD3
Dígito 5 ánodo	PD4
Dígito 6 ánodo	PD5
Dígito 7 ánodo	PD6
Dígito 8 ánodo	PD7



3.2.4 7 Segmentos Exterior

Para estos no encontramos módulos comerciales, por lo que serían contruidos a partir de tiras de LED para los diferentes segmentos y utilizamos un módulo expensor I2C MCP23017 (También lo usamos para los leds exteriores), que permite controlar hasta 16 líneas digitales mediante un bus de solo dos hilos, reduciendo los pines necesarios. Sin embargo, las salidas del MCP23017 no son capaces de manejar la corriente de las tiras LED, por lo que en cada salida controla la puerta de un MOSFET de potencia, encargado de conmutar cada segmento. Así el MCP23017 actúa únicamente como interfaz lógica, mientras que los MOSFET alimenta cada segmento LED.

La conexión del expensor seria:

Pin Módulo	Funcion	Conexión
SDA	Datos	PB8 (I2C1_SDA)
SCL	Serial Clock	PB9 (I2C1_SCL)

3.2.5 Botones Chofer

Botón	Conexión
F1	PA0
F2	PA1
F3	PA2
F4	PA3
ACEPTAR	PA4
CANCELAR	PA5

3.2.6 Botones Destino

Botón	Conexión
SM	PC8
MO	PC9
CO	PC10
AG	PC11

Los botones del sistema se conectan a pines GPIO configurados como entrada con pull-up. En reposo el pin se mantiene en nivel alto, y al presionar el botón se conecta a GND. Esta topología es robusta y evita estados flotantes.

Para garantizar un funcionamiento confiable en un entorno vehicular, cada botón puede incorporar un capacitor de aprox 100 nF para mejora del filtrado, mientras que el rebote mecánico es gestionado por software mediante técnicas de debouncing (Aunque también podría optarse por una solución por hardware).

Los botones del chofer se conectan a PA0–PA5 y los botones de destino a PC8–PC11, sin interferir con los pines usados para multiplexar los displays en PC0–PC7.

3.2.7 LEDs

Los cuatro indicadores al frente de la unidad se alimentarán desde la fuente con 5V y serán LEDs de alto brillo SMD 2835/5050 (15–30 mA). Cada LED se conecta en serie con una resistencia calculada para fijar la corriente y se conmuta por un MOSFET. Las puertas de los MOSFET son activadas por salidas del MCP23017, de modo que los MOSFETs manejan la corriente de potencia. Se incorpora un resistor de puerta (100 Ω) y un pulldown (100 k Ω) por MOSFET.

Pin Módulo	Funcion	Conexión
SDA	Datos	PB8 (I2C1_SDA)
SCL	Serial Clock	PB9 (I2C1_SCL)

3.2.8 GPS

El módulo GPS opera como un receptor pasivo de radiofrecuencia sintonizado a la banda L1; esta banda es la señal civil estándar del sistema GPS, transporta el código adquisición y el mensaje de navegación que el propio módulo utiliza internamente para calcular la posición, la cual es posteriormente transmitida al microcontrolador. Se eligió un módulo receptor compatible con esta banda debido a su amplia disponibilidad, bajo costo y precisión suficiente para aplicaciones de logística vehicular. Su función principal es la demodulación de señales provenientes de la constelación de satélites NAVSTAR.

El cálculo de la posición es realizado por el módulo mediante el método de trilateración; para obtener una solución de navegación bidimensional (latitud y longitud), el sistema requiere, matemáticamente, la intersección de esferas de señal de un mínimo de tres satélites.

La transmisión de información puede darse incluso aunque no haya realizado el cálculo o no haya sintonizado la señal de tres satélites, por eso se usa una bandera de validación denominada FIX. Esta bandera es un estado lógico que indica que el receptor ha logrado decodificar las efemérides, sincronizar su reloj interno y resolver las ecuaciones de trilateración con un error aceptable.

- **Estado "No FIX":** Al inicio o en zonas sin cobertura (ej. túneles), el módulo no puede garantizar la ubicación.
- **Estado "FIX Válido":** El módulo garantiza que las coordenadas entregadas corresponden a la realidad física actual.

Si el sistema ignora esta bandera, corre el riesgo de procesar coordenadas nulas, lo que llevaría a errores críticos en la determinación del destino del colectivo.

La comunicación entre el módulo y el micro se realiza mediante una interfaz UART debido a que el módulo integra de forma nativa conexiones para esta interfaz:

El módulo GPS actúa como un transmisor, enviando ráfagas de tramas NMEA a una frecuencia de actualización de 1Hz. El microcontrolador utiliza una interrupción de recepción que se dispara byte a byte para capturar el flujo de datos entrante.

La información se transmite codificada en formato ASCII siguiendo el estándar NMEA 0183. Este protocolo organiza los datos en "sentencias" o líneas de texto legibles.

Para este proyecto, se ha seleccionado la sentencia \$GPRMC (Recommended Minimum Specific), ya que consolida la información esencial en una única estructura lineal.

Ejemplo de la estructura de la trama \$GPRMC:

```
$GPRMC,142530.00,A,2649.883,S,06513.516,W,015.5,090.0,251224,,A*6B
```

Dato en la trama	Nombre del campo	Descripción Técnica
\$GPRMC	Header	Inicio de sentencia (\$), ID de constelación (GP=GPS) y tipo de mensaje (RMC).
142530.00	Hora (UTC)	Son las 14:25:30. Recuerda restar 3 horas para Argentina (11:25:30).
A	Estado (FIX)	A = Active (Válido). Si fuera 'V' (Void), los datos siguientes serían basura.

2649.883	Latitud	26 grados, 49.883 minutos. (Formato DDMM.MMM).
S	Hemisferio Lat.	Sur. (Significa que la latitud es negativa: -26...).
06513.516	Longitud	065 grados, 13.516 minutos.
W	Hemisferio Long.	West (Oeste). (Significa que la longitud es negativa: -65...).
015.5	Velocidad	Velocidad sobre tierra en Nudos. (Aprox 28.7 km/h).
090.0	Curso (Track)	Dirección de movimiento en grados 90°= Viajando hacia el Este
251224	Fecha	25 de diciembre de 2024. (Formato dd-mm-aa).
„	Declinación Mag.	(Vacío en la mayoría de módulos como el NEO-6M).
*6B	Checksum	Suma de verificación hexadecimal para comprobar errores de ruido.

Conexión del GPS:

Pin Módulo	Funcion	Indispensable	Conexión
VCC	+3.3V	Si	Alimentación
GND	Tierra	Si	Tierra
TX	Salida serial del gps	Si	PA10 (USART0_RX)
RX	Entrada serial del gps	Si	PA9 (USART0_TX) <i>(no se usa, pero igual se conecta)</i>
PPS	Salida de 1Hz	Opcional	-

3.2.9 Memorias

Para garantizar la integridad de los datos y la eficiencia operativa del sistema, basamos la estrategia de hardware en la segmentación física de los datos según su volatilidad y frecuencia de acceso, aprovechando las características distintivas de las dos tecnologías de almacenamiento interno disponibles: Flash y SRAM.

3.2.9.1 Memoria Flash

Desde el punto de vista del diseño de hardware, esta memoria presenta tres características críticas que condicionan su uso en el proyecto:

- **Persistencia:** Es una memoria de tipo no volátil, lo que garantiza la retención de datos en ausencia de suministro eléctrico.
- **Acceso de Lectura:** Está optimizada para la recuperación rápida de instrucciones y constantes a través del bus de código (I-Code/D-Code).
- **Restricciones de Escritura (Endurance):** A diferencia de la RAM, la tecnología Flash posee una vida útil limitada en cuanto a ciclos de borrado/escritura (típicamente garantizada hasta 100.000 ciclos). Además, el proceso de escritura requiere el borrado previo de páginas completas y tiempos de latencia del orden de los milisegundos, lo que la hace inviable para el almacenamiento de variables de alta dinámica o cambios en tiempo real durante el recorrido.

Nota: Esta restricción física es el factor determinante para decidir alojar aquí únicamente datos inmutables como la Matriz de Tarifas.

CiudadOrigen	CiudadDestino	Precio (ARS\$)
San Miguel de Tucumán	Monteros	\$3,310.00
San Miguel de Tucumán	Concepción	\$4,580.00
San Miguel de Tucumán	Aguilares	\$5,010.00
Monteros	San Miguel de Tucumán	\$3,310.00
Monteros	Concepción	\$2,450.00
Monteros	Aguilares	\$3,300.00
Monteros	San Miguel de Tucumán	\$4,580.00
Concepción	Monteros	\$2,450.00
Concepción	Aguilares	\$1,580.00
Aguilares	San Miguel de Tucumán	\$5,010.00
Aguilares	Monteros	\$3,300.00
Aguilares	Concepción	\$1,580.00

Interfaz de Conexión	Función
Bus I-Code (Instruction)	Lectura de Instrucciones: Por donde la CPU "lee" el programa compilado (.text).
Bus D-Code (Data)	Lectura de Constantes: Por donde la CPU lee la Matriz de Tarifas (.rodata).

3.2.9.2 Memoria SRAM

Las especificaciones técnicas consideradas para el diseño son:

- **Acceso Aleatorio de Baja Latencia:** La SRAM permite operaciones de lectura y escritura en un único ciclo de reloj, fundamental para atender interrupciones críticas (como la llegada de datos GPS o tramas RFID) sin detener el procesador.
- **Ciclos de Escritura:** Físicamente, las celdas de SRAM no sufren degradación por reescritura. Esta característica es vital para estructuras de datos transaccionales que se actualizan cientos de veces por hora.
- **Volatilidad:** Al depender de la alimentación eléctrica para retener los estados lógicos, cualquier dato almacenado aquí se perderá ante un reinicio o corte de energía, requiriendo una reinicialización por software en cada arranque.

Nota: Debido a su resistencia al desgaste y velocidad, este es el soporte físico seleccionado para gestionar la Tabla de Pasajeros, la Tabla de Contadores de Pasajeros y los estados lógicos del viaje.

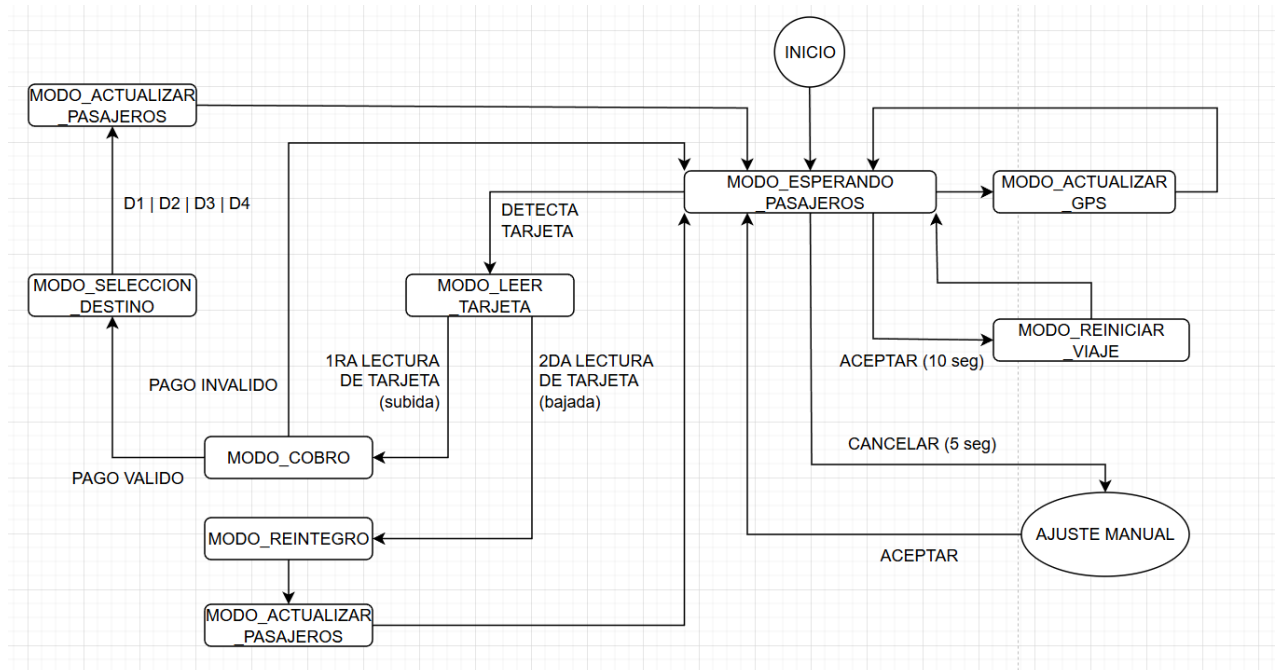
TarjetaID	Origen
0x3A81C295	San Miguel de Tucumán
0xB4F2E1A0	San Miguel de Tucumán
0x91C56D33	San Miguel de Tucumán
0x04782EA1	Monteros
0x80521F4B	Concepción
0xF2A933CC	Aguilares
0x040A15D1	Concepción
0x04C80A11	Monteros

Interfaz de Conexión (Bus Interno)	Tipo de Acceso	Función en el Proyecto
System Bus (S-Bus)	Lectura / Escritura	Variables Dinámicas: Almacenamiento de la Tabla de Pasajeros y Stack (.bss / .data).
Bus Matrix	Control	Gestiona el acceso simultáneo.

4. Diseño de Software

4.1 Funcionamiento General

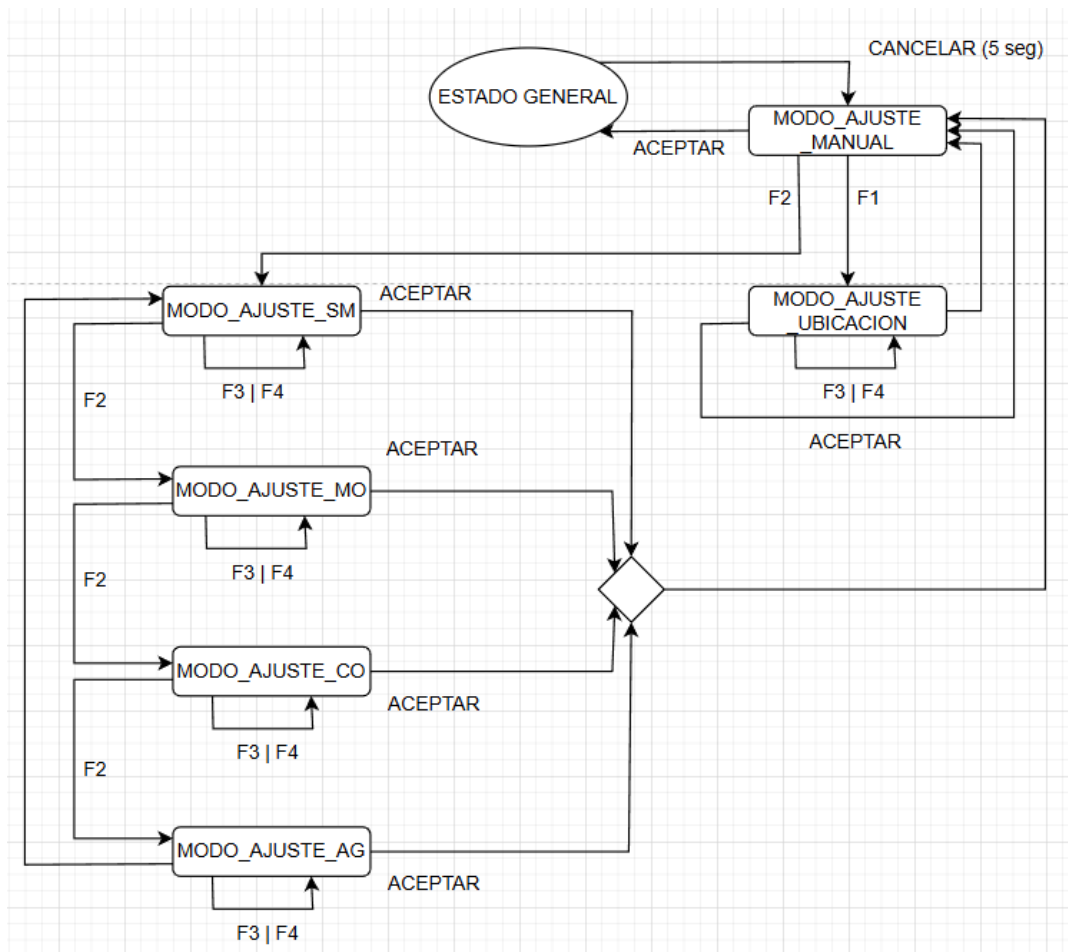
4.1.1 Diagrama de funcionamiento general



El diagrama modela al sistema como una máquina de estados finitos cuyo estado principal es **MODO_ESPERANDO_PASAJEROS**, al que siempre se regresa. Cuando el lector detecta una tarjeta se pasa a **MODO_LEER_TARJETA**, donde se guarda su ID y se decide si es una primera lectura (sube) o una segunda (baja).

Si es la primera lectura, el sistema entra en **MODO_COBRO** para debitar la tarifa máxima; si el pago es válido, pasa a **MODO_SELECCION_DESTINO** y luego a **MODO_ACTUALIZAR_PASAJEROS**, actualizando la tabla interna antes de volver al modo de espera. Por otro lado, si es segunda lectura, se ingresa a **MODO_REINTEGRO**, donde se calcula la tarifa real y el monto a devolver, y luego se actualiza la tabla en **MODO_ACTUALIZAR_PASAJEROS**.

Como modos auxiliares, desde el modo de espera se accede periódicamente a **MODO_ACTUALIZAR_GPS** para determinar la ciudad actual y el próximo destino automático; una pulsación larga de "Aceptar" lleva a **MODO_REINICIAR_VIAJE** (reseteo de contadores y tabla), y una pulsación larga de "Cancelar" entra en **AJUSTE_MANUAL**, donde se modifican contadores y posición manual antes de volver al modo de espera.



En MODULO AJUSTE MANUAL el chofer usa F1 y F2 para elegir entre los ajustes:
 MODULO AJUSTE UBICACION: con F3 y F4 fija manualmente la ciudad actual y, al aceptar, se corrige el próximo destino mostrado.

MODULO AJUSTE SM/MO/CO/AG: con F3 y F4 aumenta o disminuye los contadores de pasajeros de cada ciudad y con F2 pasa al siguiente contador.

En cualquier caso, al presionar Aceptar se guardan los cambios y se vuelve al modo general, permitiendo corregir situaciones especiales sin frenar el funcionamiento normal del sistema.

4.1.2 Tabla de funcionamiento

Resume los modos de funcionamiento en los que puede encontrarse el sistema. Cada modo se identifica con un nombre simbólico y un valor numérico, que se utiliza en el programa principal como índice de la máquina de estados. De esta manera se separan claramente las distintas etapas del proceso.

Nombre	Valor	Descripción
MODO_ESPERANDO_PASAJEROS	0	Estado base del sistema. Se muestran asientos y próximo destino. Espera tarjeta o teclas del chofer.
MODO_COBRO	1	El sistema gestiona el cobro de la tarifa máxima en la 1ra pasada de tarjeta (subida), hablando con el dispositivo de cobro.
MODO_SELECCION	2	Luego de un cobro válido en la subida, el sistema espera que se elija el destino

_DESTINO		del pasajero.
MODO_REINTEGRO	3	El sistema calcula la tarifa real según ciudad_origen y ciudad_actual y pide al dispositivo de cobro el reintegro (en ciertos casos será \$0).
MODO_AJUSTE_MANUAL	4	El chófer ingresa al menú de ajuste manual para corregir contadores de pasajeros por ciudad y/o la ubicación actual y próximo destino.
MODO_AJUSTE_SM	5	Ajuste manual realizado por el chófer para ajustar contadores de pasajeros cuyo destino es San Miguel de Tucumán.
MODO_AJUSTE_MO	6	Ajuste manual realizado por el chófer para ajustar contadores de pasajeros cuyo destino es Monteros.
MODO_AJUSTE_CO	7	Ajuste manual realizado por el chófer para ajustar contadores de pasajeros cuyo destino es Concepción.
MODO_AJUSTE_AG	8	Ajuste manual realizado por el chófer para ajustar contadores de pasajeros cuyo destino es Aguilares.
MODO_ACTUALIZAR_PASAJEROS	9	Actualiza el listado de pasajeros accediendo a memoria.
MODO_AJUSTE_UBICACION	10	Ajuste manual realizado por el chófer para corregir la el próximo destino mostrados en el panel.
MODO_ACTUALIZAR_GPS	11	Si se detecta un cambio de ciudad, actualiza proximo_destino y refresca la información mostrada al chofer y a los pasajeros.
MODO_REINICIAR_VIAJE	12	Al finalizar el recorrido, reestablece la tabla de pasajeros en viaje, contadores por ciudad y asientos a sus valores iniciales (sin cambiar ciudad_actual).
MODO_LEER_TARJETA	13	Gestiona la detección y lectura de la tarjeta. Obtiene el id_tarjeta y decide si es 1ra lectura (subida) o 2da lectura (bajada).

4.2 Estructura de datos

4.2.1 Tabla de estructura de datos

Etiqueta	Tamaño	Descripción
Modo	1 Byte	Modo de funcionamiento actual del sistema (máquina de estados principal).
Refresco	1 Byte	Índice del dígito actual en los displays de 7 segmentos multiplexados.
DigitosFrontal	4 Bytes	Valores de los dígitos del display frontal (próximo destino y asientos disponibles).
DigitosPanel	4 Bytes	Valores de los dígitos del display del chofer (pasajeros por destino).
Divisor	2 Bytes	Contador de interrupciones de timer utilizado como base de tiempos (ms → segundos).
Tiempo	2 Bytes	Contador de segundos del sistema (temporizadores de 5 s, 10 s, timeout, etc.).
CiudadActual	1 Byte	Ciudad en la que se encuentra actualmente el colectivo.
CiudadOrigen	1 Byte	Ciudad de origen del pasajero actual (leída desde la tabla de pasajeros para el reintegro).
ProximoDestino	1 Byte	Próxima ciudad del recorrido que se muestra como “próximo destino”.
AsientosDisponibles	1 Byte	Cantidad de asientos disponibles mostrada por el sistema (derivada de los contadores por ciudad).
ContadorSM	1 Byte	Pasajeros cuyo destino es San Miguel de Tucumán.
ContadorMO	1 Byte	Pasajeros cuyo destino es Monteros.
ContadorCO	1 Byte	Pasajeros cuyo destino es Concepción.

ContadorAG	1 Byte	Pasajeros cuyo destino es Aguilares.
Tarjeta	4 Bytes	Identificador de la tarjeta leída actualmente por el sistema.
EstadoTarjeta	1 Byte	Estado de la tarjeta (sin viaje, en viaje, bajando, error, etc.).
ExisteTarjeta	1 Byte	Bandera que indica si la tarjeta leída se encontró en la tabla de pasajeros (1 = sí, 0 = no).
IndicePasajero	1 Byte	Posición en el arreglo Pasajeros[] donde se encuentra la tarjeta actual (válido si ExisteTarjeta=1).
TarifaMaxima	2 Bytes	Tarifa máxima cobrada al momento de la subida (viaje completo).
TarifaReal	2 Bytes	Tarifa real del viaje calculada al momento de la bajada (según CiudadOrigen y CiudadActual).
Reintegro	2 Bytes	Monto a reintegrar al pasajero (puede ser 0).
Intento	1 Byte	Contador de reintentos al comunicarse con el dispositivo de cobro en la rutina de reintegro/pago.
Tecla Presionada	1 Byte	Código de la tecla actualmente pulsada (F1–F4, ACEPTAR, CANCELAR, ninguna).
Duración	1 Byte	Duración de la pulsación de la tecla actual (usada para detectar pulsaciones largas de 5 s y 10 s).
Actuales	1 Byte	Valor actual de las entradas de teclado (máscara de bits leída del puerto).
Anterior	1 Byte	Valor anterior de las entradas de teclado (para detectar flancos y cambios).
Inactividad	1 Byte	Cantidad de segundos sin actividad en las teclas (timeout de selección / ajuste).
GpsNuevo	1 Byte	Bandera que indica si se recibió un nuevo mensaje válido desde el módulo GPS (1 = nuevo dato).
Fix	1 Byte	Bandera de fijación del GPS (1 = posición válida, 0 = sin FIX).
LatitudGPS	4 Bytes	Última latitud recibida desde el módulo GPS, usada para determinar la ciudad detectada.
CiudadDetectada	1 Byte	Ciudad estimada a partir de la última lectura de GPS (previa a actualizar CiudadActual).
SentidoViaje	1 Byte	Sentido actual del recorrido (por ejemplo 0 = SM a AG, 1 = AG a SM).
LedSM	1 Byte	Estado del LED que indica “próximo destino San Miguel de Tucumán”.
LedMO	1 Byte	Estado del LED que indica “próximo destino Monteros”.
LedCO	1 Byte	Estado del LED que indica “próximo destino Concepción”.
LedAG	1 Byte	Estado del LED que indica “próximo destino Aguilares”.
BufferLCD	80 Bytes	Memoria intermedia para el contenido de la pantalla LCD 20×4 (texto a mostrar en cada posición).
Pasajeros[]	~8×43 Bytes	Tabla en memoria volátil con los datos de los pasajeros en viaje (tarjeta, ciudad de origen, estado).

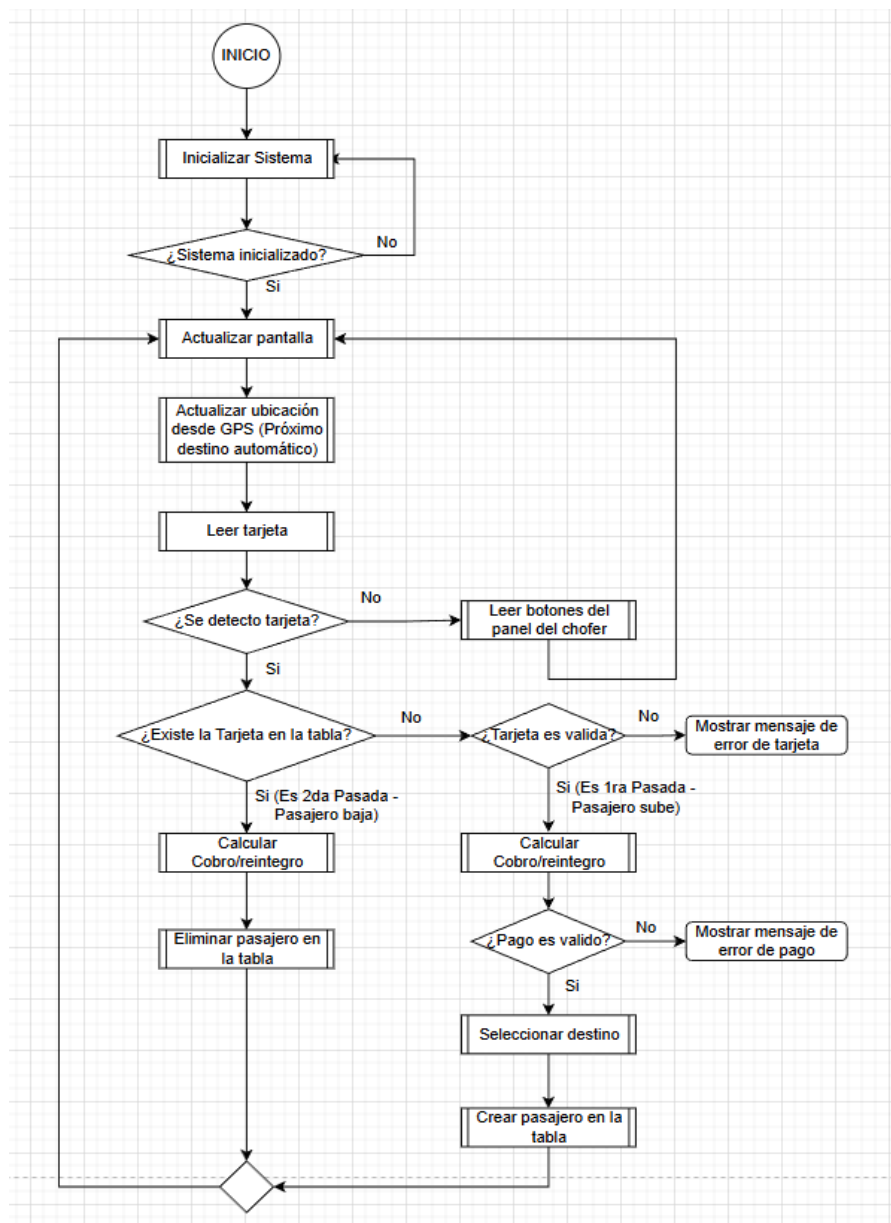
La tabla describe las variables en RAM sobre las que trabaja el sistema, agrupadas en cinco bloques lógicos:

- **Grupo 1 – Estado general del viaje:** modo actual, ciudad actual y próximo destino, contadores de pasajeros por ciudad, asientos disponibles y variables de tiempo; reflejan en todo momento la situación del colectivo y habilitan los temporizadores.
- **Grupo 2 – Pasajeros y cobro:** datos de la tarjeta y su estado, índice del pasajero, tabla de pasajeros, ciudad de origen, tarifas y reintegro; permiten saber si es primera o segunda pasada y calcular los montos a debitar o devolver.

- **Grupo 3 – Teclado y chofer:** variables para tecla actual, duración y estados presente/anterior; sirven para antirrebote y detección de pulsaciones largas de 5 y 10 segundos.
- **Grupo 4 – GPS y ubicación automática:** flags y datos de GPS, ciudad detectada y sentido de viaje; se usan para determinar la ciudad actual, validar descensos y fijar el próximo destino.
- **Grupo 5 – Salidas de visualización:** control del refresco, dígitos de displays, LEDs de destino y buffer de la LCD; garantizan que el modelo interno se vea correctamente en los dispositivos de salida.

4.3 Diagramas de flujos de las rutinas

4.3.1 Diagrama General de Software



El diagrama general de software muestra un ciclo principal donde, tras Inicializar Sistema, el microcontrolador actualiza la pantalla y la ubicación por GPS en cada vuelta. Luego intenta leer una tarjeta, si no hay tarjeta, lee los botones del panel del chofer y reinicia el ciclo. por otro lado si hay tarjeta, verifica si ya está en la tabla y si es válida.

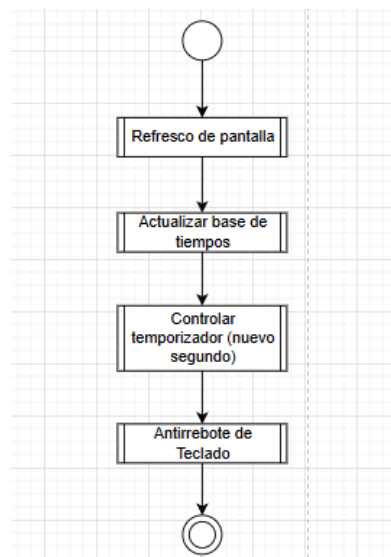
A continuación comprueba:

Segunda pasada (bajada): calcula cobro/reintegro y elimina al pasajero de la tabla.

Primera pasada (subida): si la tarjeta es inválida muestra error; si es válida calcula el cobro, consulta al dispositivo de pago y, solo con pago aprobado, permite elegir destino y crea al pasajero en la tabla.

Tras cualquiera de estos caminos, el flujo vuelve al inicio del lazo, actualiza la pantalla y queda nuevamente a la espera de tarjetas o comandos del chofer.

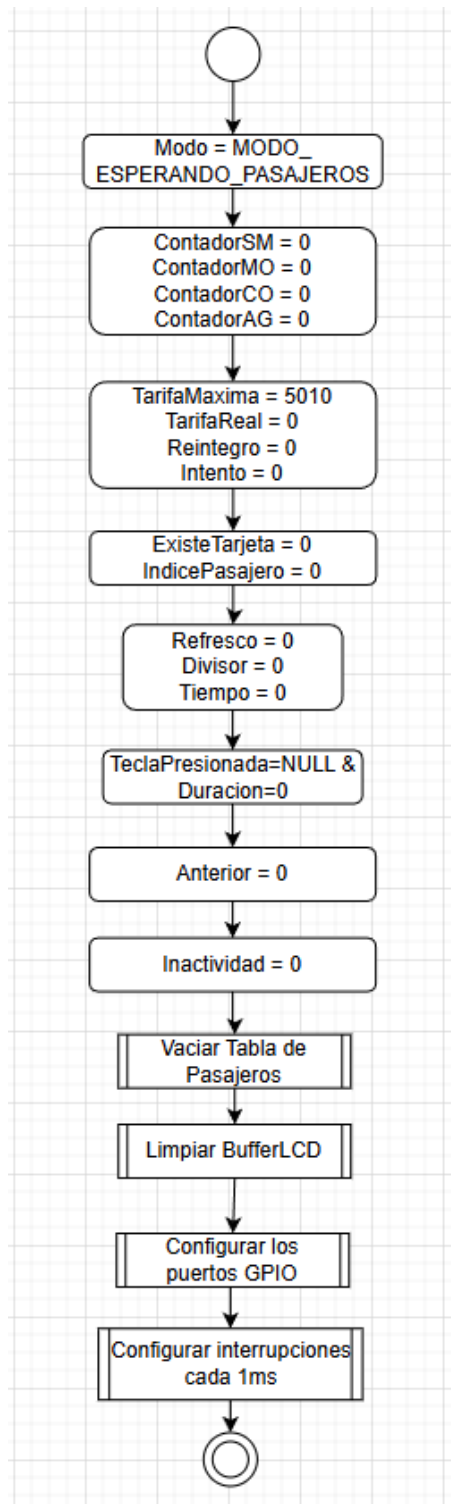
4.3.2 Diagrama General de Interrupciones del Timer



En el diagrama general de las interrupciones de timer se muestra lo que ejecuta el microcontrolador cada vez que vence el temporizador periódico (por ejemplo, cada 1 ms).

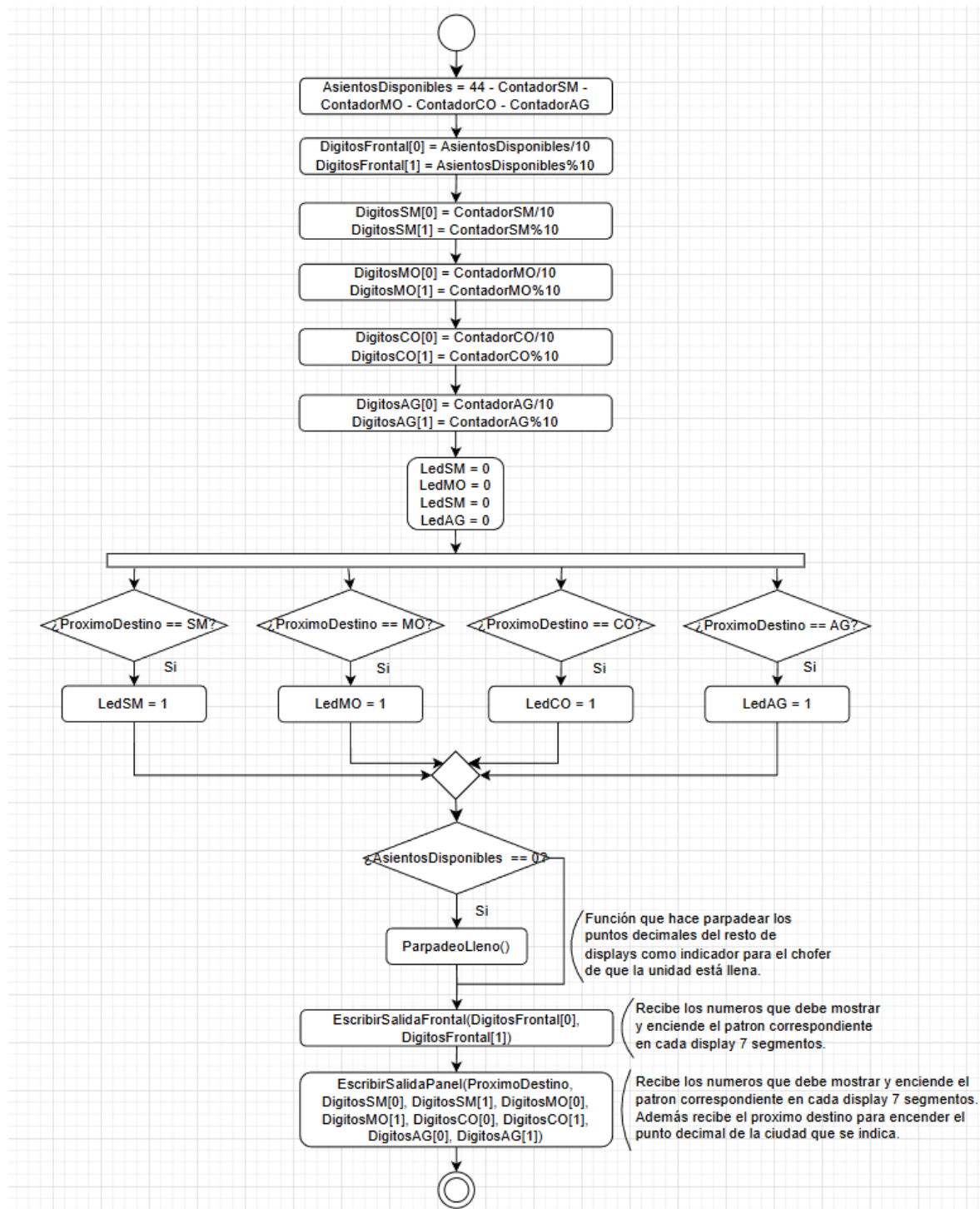
En primer lugar se realiza el refresco de pantalla, actualizando los displays multiplexados y los indicadores luminosos sin detener el programa principal. Luego se actualiza la base de tiempos. A continuación se controla el temporizador de “nuevo segundo”: cuando la cantidad de ticks acumulados alcanza el valor prefijado. Finalmente se ejecuta el antirrebote de teclado. De esta manera, se concentran todas las tareas periódicas de tiempo real del sistema.

4.3.3 Diagrama Rutina de Inicialización



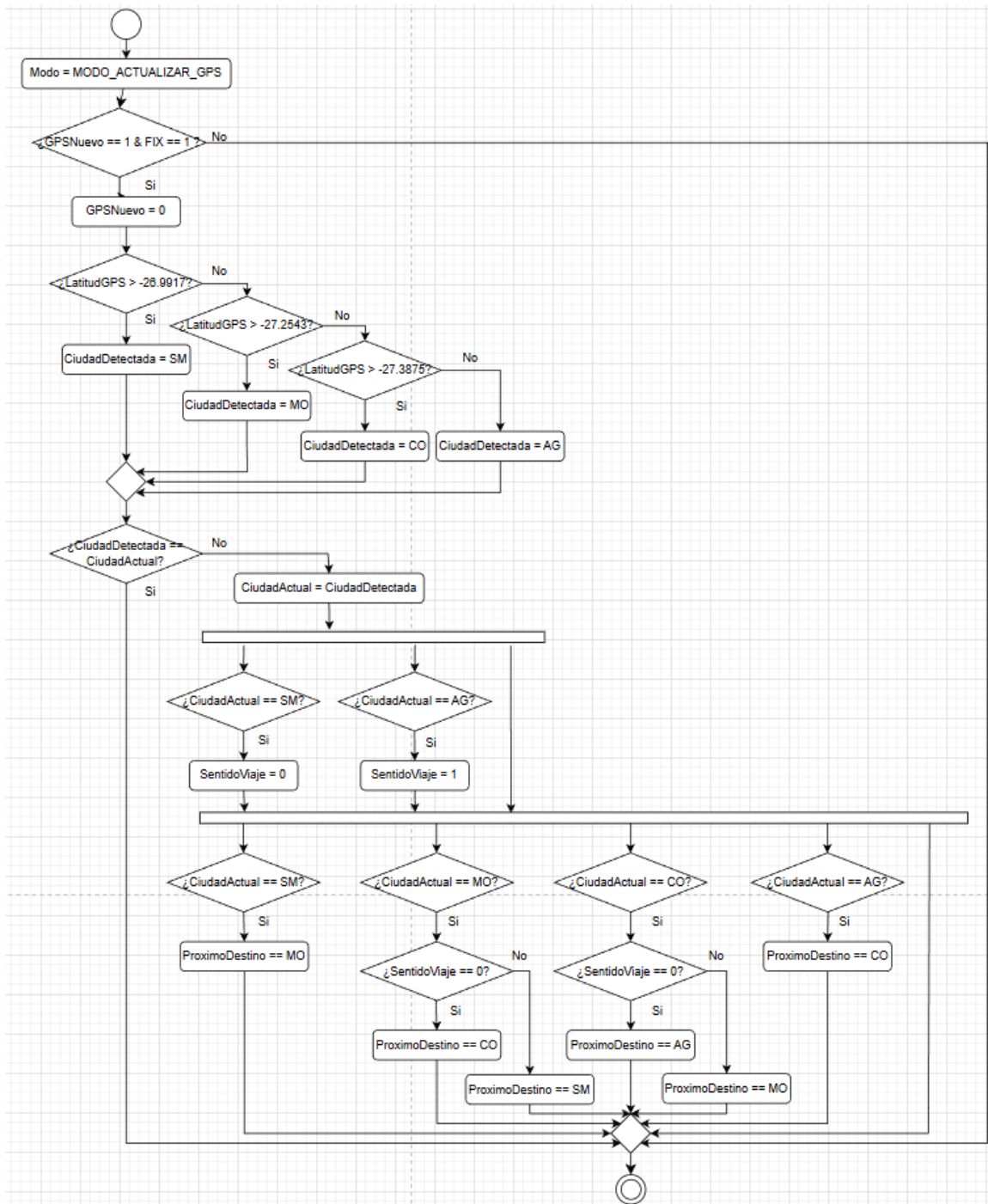
La rutina de inicialización se ejecuta al encender el sistema y deja todo en un estado conocido: pone el sistema en MODO_ESPERANDO_PASAJEROS, pone en cero los contadores por ciudad, configura las variables de cobro (TarifaMaxima, TarifaReal, Reintegro, Intento) y limpia las banderas y los contadores de tiempo. Luego inicializa las variables del teclado del chofer, vacía la tabla de pasajeros, borra el BufferLCD y finalmente configura los puertos GPIO y el temporizador para generar interrupciones cada 1 ms.

4.3.4 Diagrama Rutina de Actualizar Pantalla



La rutina ActualizarPantalla convierte el estado interno en información visible. Calcula los asientos disponibles (44 menos la suma de los contadores por ciudad), descompone ese valor en decenas y unidades y lo carga en los dígitos del display frontal, haciendo algo similar para cada contador de ciudad. Enciende solo el LED del próximo destino, verifica si la unidad está llena para indicarlo con ParpadeoLleno y luego llama a EscribirSalidaFrontal y EscribirSalidaPanel, que actualizan los patrones de los displays de 7 segmentos.

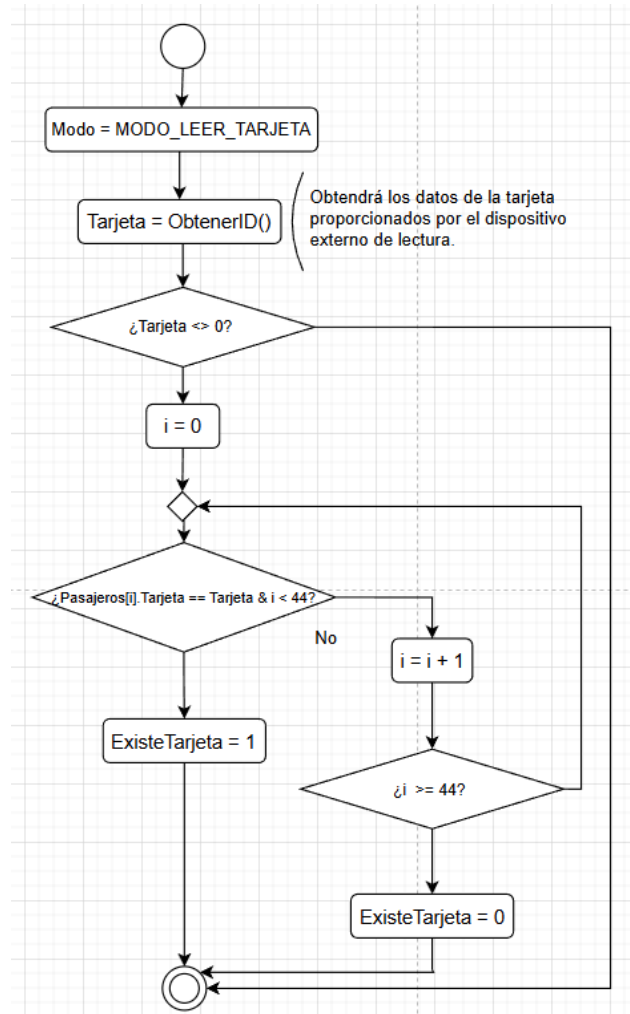
4.3.5 Diagrama Rutina de Actualizar Ubicación desde GPS



La rutina de actualización desde GPS solo actúa cuando llega un dato nuevo y válido (GpsNuevo=1 y FIX=1). Entonces borra GpsNuevo, clasifica la latitud en uno de los cuatro rangos de ciudades (SM, MO, CO, AG) y actualiza CiudadDetectada y, si cambió, también CiudadActual y el sentido del viaje. Con esos datos calcula el próximo destino.

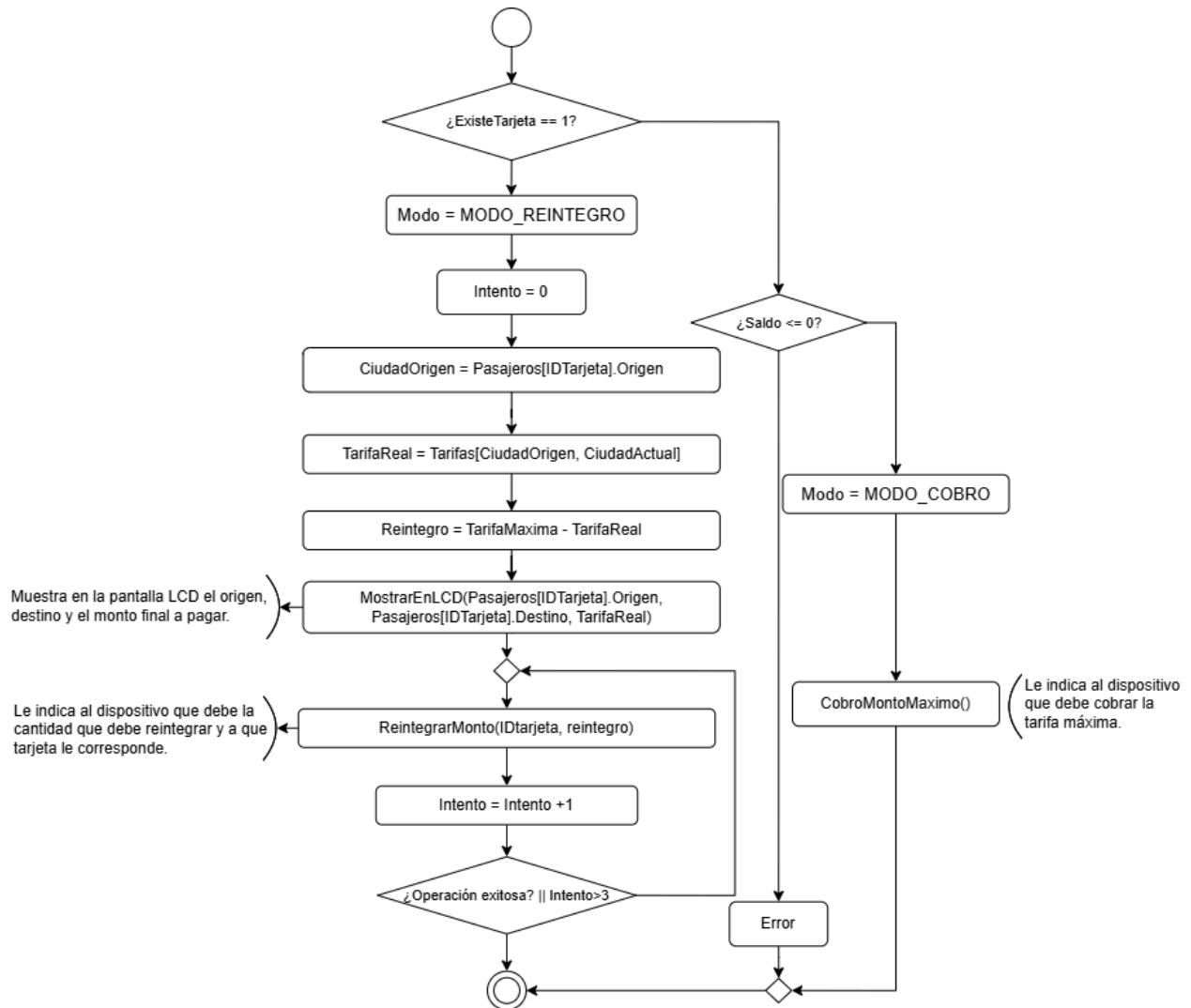
Además, una vez que se supera la latitud de una ciudad en el sentido de marcha, no se vuelve a una ciudad anterior, evitando cambios erráticos en la detección.

4.3.6 Diagrama Rutina de Leer Tarjeta



La rutina LeerTarjeta obtiene el ID de la tarjeta y verifica si el pasajero ya está en viaje. Llama a ObtenerID() y guarda el resultado en Tarjeta; si vale 0, termina sin cambios. Si es distinto de 0, recorre la tabla Pasajeros[] (hasta 44 posiciones) comparando cada Pasajeros[i].Tarjeta con el ID leído. Si encuentra coincidencia, fija ExisteTarjeta = 1 y guarda el índice, indicando segunda pasada. Si no encuentra ninguna, deja ExisteTarjeta = 0, señalando que se trata de una tarjeta nueva (pasajero aún no registrado).

4.3.7 Diagrama Rutina de Calcular Cobro/Reintegro

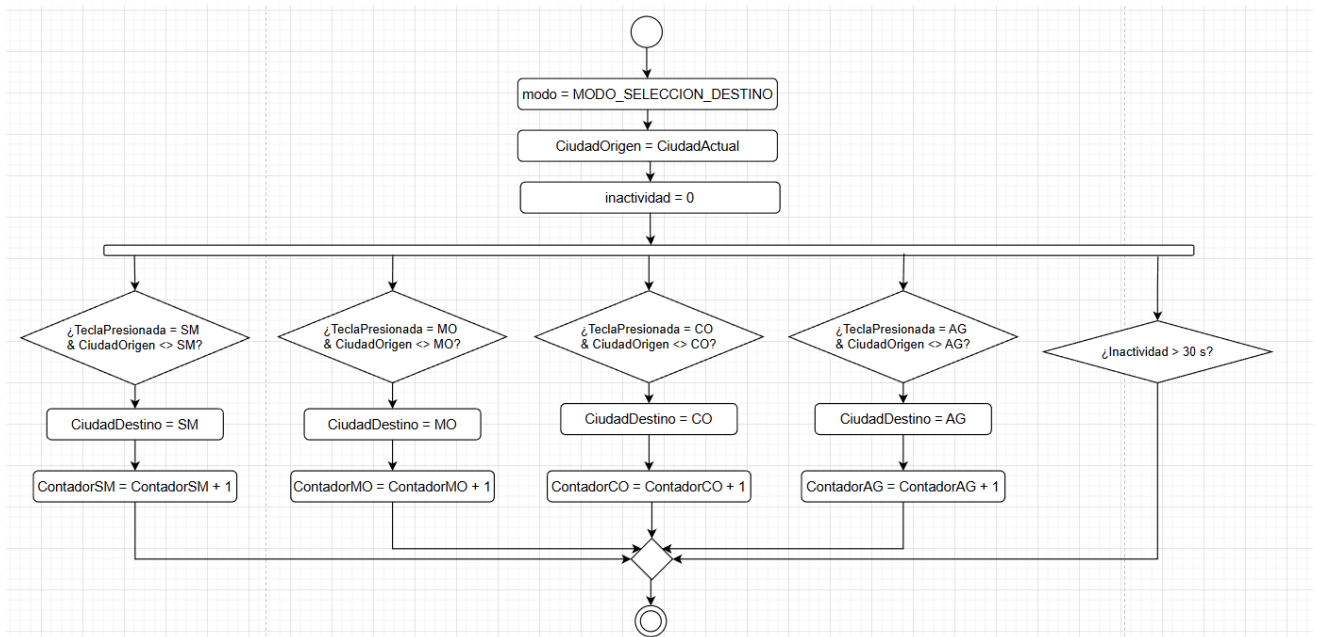


La rutina Calcular Cobro/Reintegro define cómo operar con el dispositivo de pago según ExisteTarjeta:

Primera pasada (tarjeta nueva, no está en la tabla): se pone Modo = MODO_COBRO, se verifica el saldo y, si alcanza, se llama a CobroMontoMaximo() para debitar la tarifa máxima; si no hay saldo suficiente, se muestra en la LCD un mensaje indicando que el pasajero debe bajar.

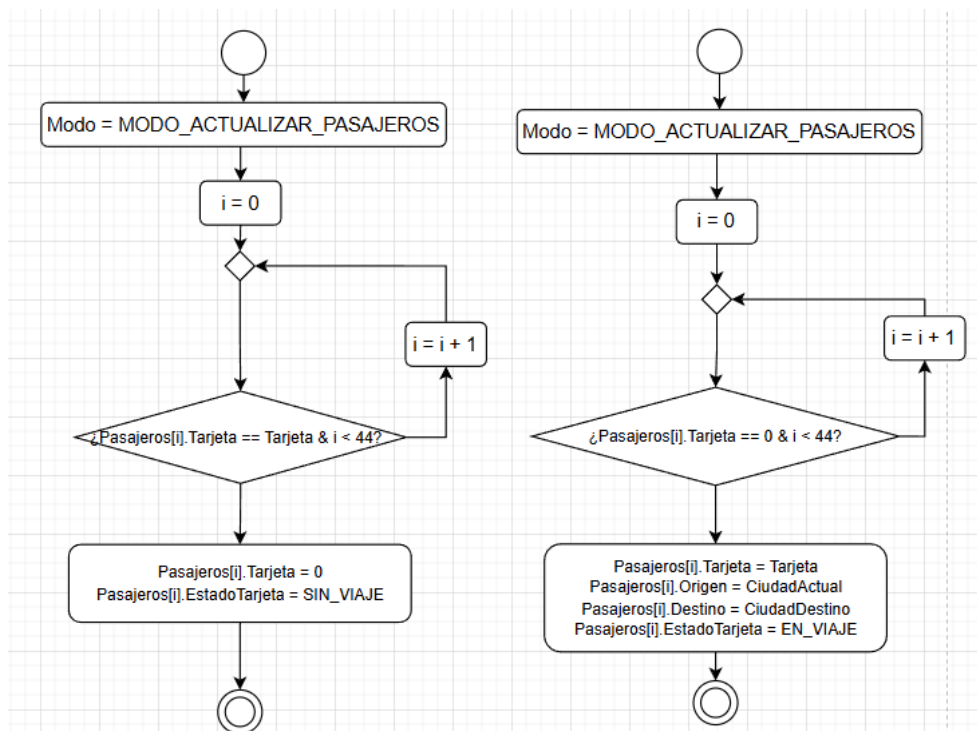
Segunda pasada (tarjeta ya registrada): se pone Modo = MODO_REINTEGRO, se obtiene la CiudadOrigen de la tabla, se calcula TarifaReal = Tarifas(CiudadOrigen, CiudadActual) y el Reintegro = TarifaMaxima - TarifaReal. Se muestra Origen, Destino y TarifaReal en la LCD y se intenta ReintegrarMonto(IDTarjeta, Reintegro) hasta 3 veces. Luego el control vuelve al programa principal para crear o eliminar al pasajero en la tabla según corresponda.

4.3.8 Diagrama Rutina de Seleccionar Destino



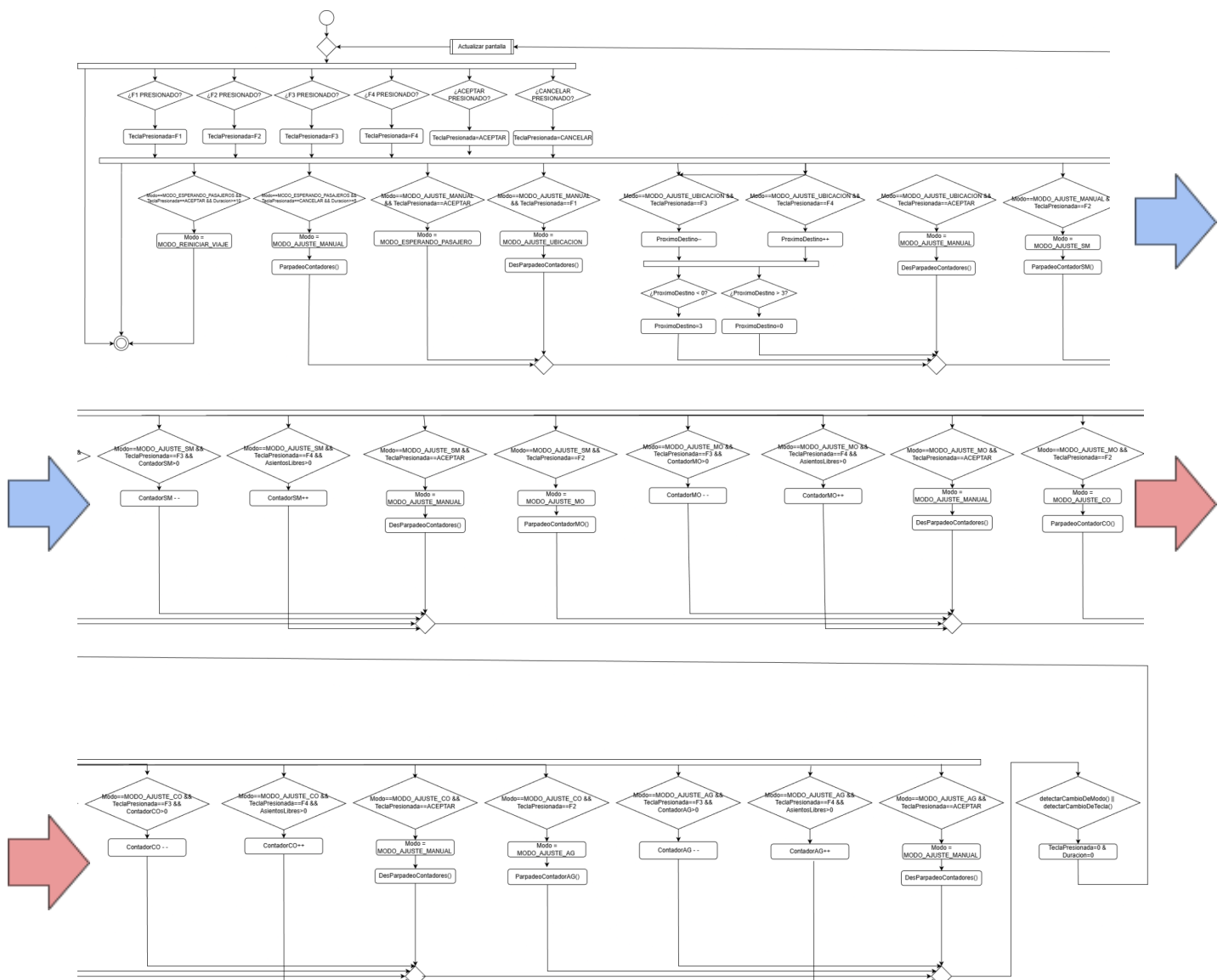
La rutina de Selección de Destino se ejecuta tras un pago válido, en MODO_SELECCION_DESTINO. Primero fija CiudadOrigen = CiudadActual y pone el temporizador de inactividad en 0. Luego espera que el chofer elija una ciudad (SM, MO, CO o AG); solo acepta la tecla si el destino es distinto del origen, en cuyo caso guarda CiudadDestino y aumenta el contador de esa ciudad. Si pasan más de 30 s sin una elección válida, se sale de la rutina por inactividad, cancela la selección y se vuelve al flujo principal.

4.3.9 Diagramas Rutinas de Crear y Eliminar Pasajero



La rutina de eliminar pasajero (bajada) recorre la tabla Pasajeros[] buscando el ID de la tarjeta; cuando lo encuentra, libera esa entrada poniendo Tarjeta = 0 y EstadoTarjeta = SIN_VIAJE, dejándola disponible para reutilizar. La rutina de crear pasajero (subida) recorre la tabla desde el inicio hasta hallar la primera posición libre (Tarjeta == 0) y allí registra al pasajero: guarda el ID de la tarjeta, la ciudad de origen, el destino elegido y pone EstadoTarjeta = EN_VIAJE. De este modo se mantiene actualizada la lista de pasajeros en viaje con su origen y destino.

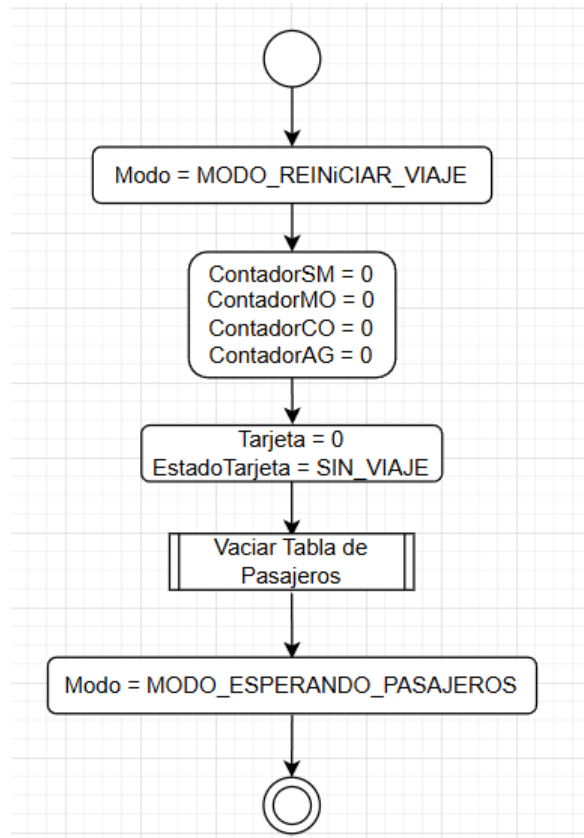
4.3.10 Diagrama Rutina de Leer Botones del Panel del Chofer



La rutina Leer botones del panel del chofer toma las teclas físicas (F1–F4, ACEPTAR, CANCELAR) y, según el modo actual y la duración de la pulsación (corta o larga, medida en la interrupción de timer), las traduce en acciones de alto nivel. Así, por ejemplo, una pulsación larga de ACEPTAR en modo normal reinicia el viaje, una pulsación larga de CANCELAR sale del ajuste manual, y las teclas F1–F4 permiten entrar a los modos de ajuste manual y de ubicación.

Dentro de los modos de ajuste, las mismas teclas sirven para modificar parámetros: en MODO_AJUSTE_SM/MO/CO/AG, F3 y F4 decrementan o incrementan el contador de pasajeros de la ciudad seleccionada y ACEPTAR confirma y vuelve al menú de ajuste; en MODO_AJUSTE_UBICACION, F3/F4 recorren el ProximoDestino entre las cuatro ciudades y ACEPTAR regresa al ajuste manual. De esta forma, la rutina centraliza toda la lógica de interpretación del panel del chofer, conectando pulsaciones físicas con cambios en contadores, próximo destino y modos de operación.

4.3.11 Diagrama Rutina de Reiniciar Viaje



La rutina de reiniciar viaje se ejecuta cuando el chofer confirma el reinicio del recorrido: pone el sistema en MODO_REINICIAR_VIAJE, pone en cero todos los contadores de pasajeros por ciudad, borra los datos de la última tarjeta usada y vacía por completo la tabla de pasajeros en viaje. Luego vuelve a MODO_ESPERANDO_PASAJEROS, dejando el equipo listo para iniciar un nuevo recorrido desde cero.

3. Conclusiones

Logramos un sistema que automatiza el contado de pasajeros y el próximo destino gracias a la información brindada por el GPS; que posibilita el ajuste manual, por parte del chofer, de los contadores y la ubicación para abordar situaciones extraordinarias; Con la información que tiene el chofer puede elegir si entrar a una ciudad o no y Implica una mejora de funcionamiento con respecto al sistema actual y Implica una mejora de funcionamiento con respecto al sistema actual.

5.1 Análisis de costos

Analizando los precios a nivel local tomando como referencia los precios por unidad en mercadolibre:

Componente	Cant.	Precio unitario aprox. (ARS)	Subtotal (ARS)
Microcontrolador GD32VF103 (LQFP64)	1	12.000	12.000
Lector RFID MFRC522	1	6.000	6.000
Tarjetas RFID MIFARE	10	1.000	10.000
Módulo GPS NEO-6M	1	8.000	8.000
Expansor I ² C MCP23017	2	4.500	9.000
Display LCD 16x2 con interfaz I ² C	1	7.000	7.000
Displays 7 segmentos interiores (SC56-11GWA)	8	2.000	16.000
Display 7 segmentos exterior (tiras LED armadas)	1	20.000	20.000
LEDs indicadores de destino (alta luminosidad)	4	800	3.200
Transistores / MOSFETs / resistencias	—	—	6.000
Pulsadores (chofer + pasajeros)	—	—	4.000
Reguladores de tensión / filtros	—	—	5.000

Componente	Costo aprox. (ARS)
PCB 2 capas	15.000
Conectores, borneras, zócalos	6.000
Cables, soldadura, consumibles	5.000

El costo total aproximado para un prototipo es de **ARS 136000**. Pero cabe la pena analizar que si se envía a hacer en china más de 10 unidades los precios pueden rondar de **50 a 100 USD** por prototipo.

5.2 Mejoras futuras

- **Tickets para manejo de errores:** En el caso de ocurrir algún error, el pasajero puede reclamar con un ticket.
- **Guardado de memoria en almacenamiento extraíble:** Posibilitando auditorías y análisis estadísticos por parte de la empresa.
- **Migrar a RFID DESFire:** Mayor seguridad.
- **Cambiar interfaz del chofer por un LCD:** Mejorar la visualización y navegación en los menús.

5.3 Análisis del uso de DMA

Si bien el micro GD32VF103 cuenta con un controlador DMA que permite que un periférico transfiera datos directamente hacia o desde la memoria RAM sin que el CPU tenga que intervenir en cada dato., en este **proyecto se decidió no utilizar dicha función.**

Los periféricos empleados (lector RFID por SPI, módulo GPS por UART, displays por I2C, y botones de entrada) generan volúmenes de datos bajos y ocasionales. En particular, el módulo GPS es usado en una velocidad reducida (9600 bps), lo que permite administrar los datos que envía por interrupciones sin riesgo de pérdida de información ni sobrecarga del procesador. El resto de los periféricos no requiere transferencias continuas ni de alta velocidad.

El uso de interrupciones y polling es suficiente para que el sistema funcione, manteniendo una programación más simple. Utilizar DMA significa aumentar la complejidad de la programación, teniendo que realizar una configuración adicional de canales, manejo de buffers y control de interrupciones, sin mejorar significativamente el rendimiento del sistema.

Por otro lado el GD32VF103 dispone de una frecuencia elevada y recursos suficientes para controlar los datos del proyecto sin inconvenientes, aún con varios dispositivos de forma simultánea. Por esto, se prioriza una arquitectura de software más simple, clara y mantenible.

En todo caso el diseño del sistema queda abierto a futuras mejoras, donde usar DMA podría ser una buena opción si aumentara el caudal de datos o la complejidad de la lógica.

4. Bibliografía

- *Cátedra de Sistemas con Microprocesadores y Microcontroladores (2025), Diapositivas de clase. Clases grabadas.*
- *GD32VF103 RISC-V 32-bit MCU, User Manual Rev 1.7.*
- *Bumblebee Processor Core, Instruction Set Architecture Manual.*
- *GD32VF103 Datasheet Rev. 1.7.*
- *Página oficial de Geodatos (obtención de las coordenadas)*
<https://www.geodatos.net/>
- *Para la realización de los diagramas se utilizó: <https://app.diagrams.net/>*