

# Proyecto integrador 2019

# Lector de tarjetas para colectivo interurbano

---

Cátedra de Sistemas con Microprocesadores y Microcontroladores

Farber, Juan

Flores Wittich, Pablo José

Romano, Hugo Leandro

**Tutor: Volentini, Esteban.**



## Contenido

- I. Introducción
- II. Análisis de mercado
- III. Funcionamiento
- IV. Descripción de componentes principales
- V. Esquema general de hardware
- VI. Tarjeta MIFARE
- VII. Estructura de datos
- VIII. Diagramas de flujo
- IX. Mejoras futuras



## I. Introducción

El objetivo de este proyecto es definir y desarrollar la implementación de un sistema de lector de tarjetas, para el cobro de viajes interurbanos en un colectivo. Es un dispositivo que en los últimos años se volvió muy común en la población, su funcionamiento es sencillo y de uso diario.

Nuestro proyecto agrega mejoras a dicho sistema, ya que permite un funcionamiento más fluido que desvincula al chofer de colectivos como intermediario entre el dispositivo y el usuario, permitiéndole focalizarse en su principal tarea que es conducir y además, no maneja dinero.

Un usuario podrá realizar una consulta de saldo en el momento que desee, acercando su tarjeta al lector.

Cabe destacar el bajo consumo del sistema y su compromiso con el medio ambiente, dejando de lado las pilas convencionales para ser energizado por el propio motor del bus.

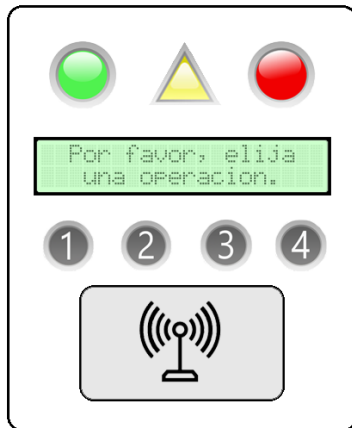


## II. Análisis de mercado

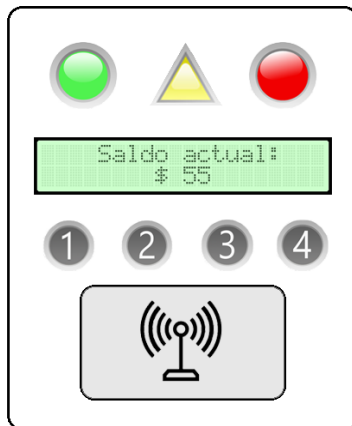
Los sistemas actuales vinculan al chofer como intermediario del usuario para escoger el destino, cuando este no es único. Esto lleva a un peligroso cambio de contexto al chofer y hace que el usuario deba lidiar con el posible error de operación ajeno.

Por otro lado, el usuario no puede realizar una consulta de saldo sin previamente abonar un viaje en los terminales existentes. Estos resultan poco interactivos y no advierten saldo bajo.

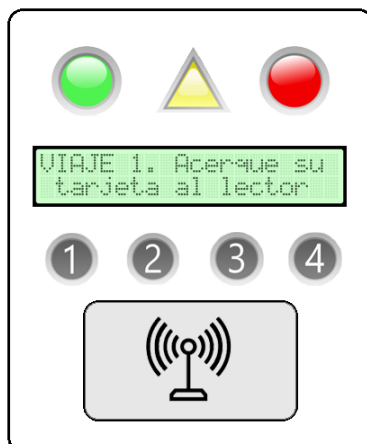
### III. Funcionamiento



Una vez que se enciende el dispositivo, el mismo muestra un mensaje de bienvenida al usuario



Al acercar una tarjeta, sin presionar alguno de los botones, se muestra por pantalla el saldo actual del mismo



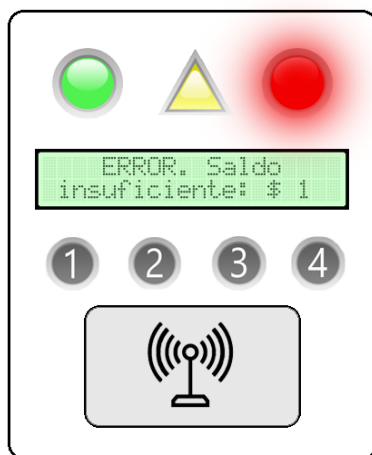
Al presionar alguno de los 4 botones, correspondiente a los distintos viajes, el dispositivo solicita al usuario que acerque su tarjeta



Si el saldo del usuario es suficiente como para abonar el viaje seleccionado, se mostrará el mensaje de éxito, junto con el saldo restante. Además, se encenderá un LED verde.



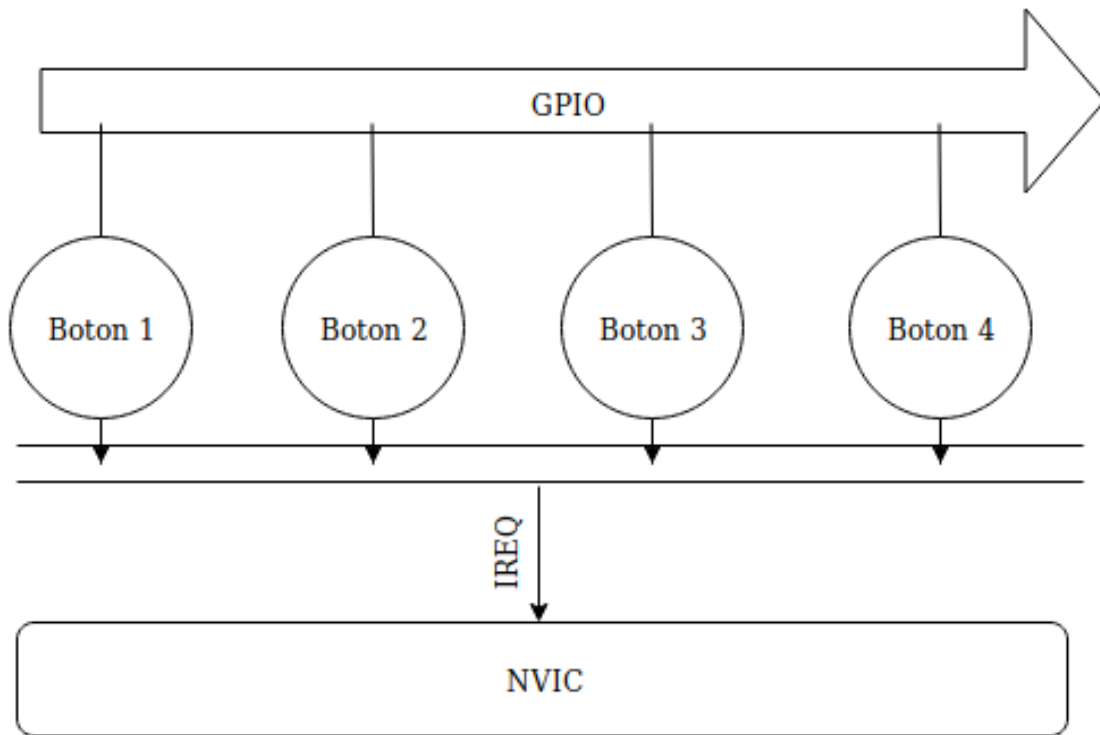
Sí el viaje fue abonado con éxito, pero el saldo restante alcanza para solo 2 viajes más (del mismo boleto abonado), se encenderá un LED amarillo, para que el usuario tenga en cuenta esta situación para el futuro.



Finalmente, si el saldo del usuario no es el suficiente para abonar el viaje, se muestra en pantalla el mensaje de error junto con su saldo actual, y se enciende el LED rojo.

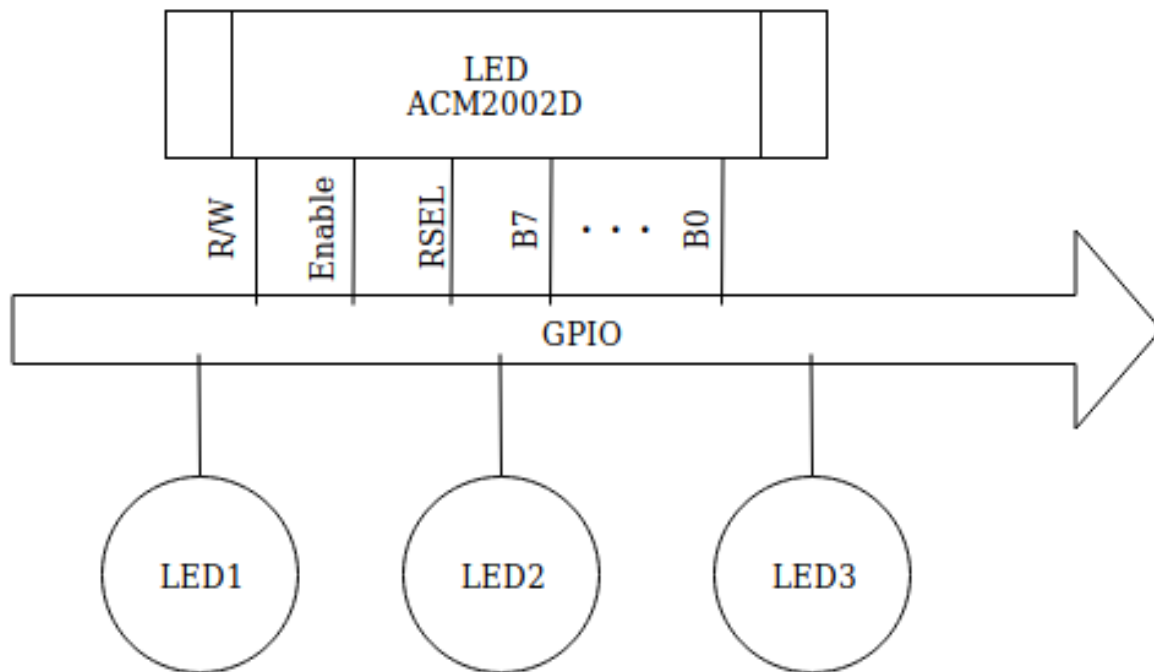
## IV. Descripción de componentes principales

### A. Botones



Como se muestra en la figura todos los botones están conectados a la misma línea de interrupción y se configuran como GPIO de entrada.

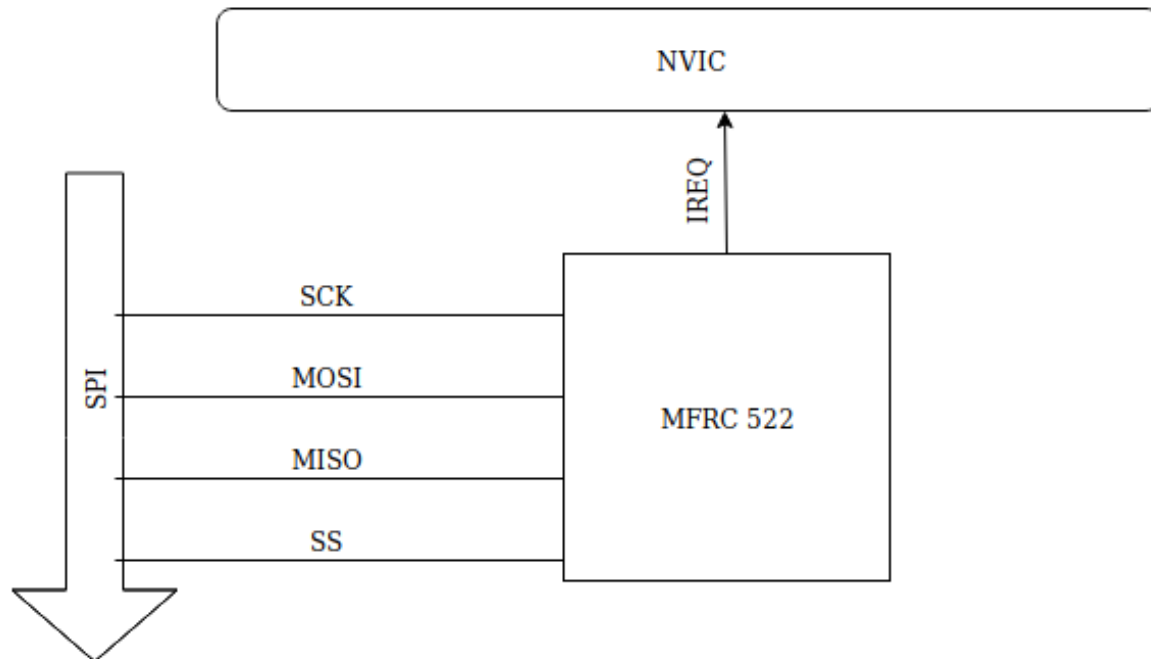
## B. Display LED



Para hacer funcionar las luces y la pantalla led, serán conectadas a terminales configurados como GPIO de salida.



## C. Módulo MFRC 522

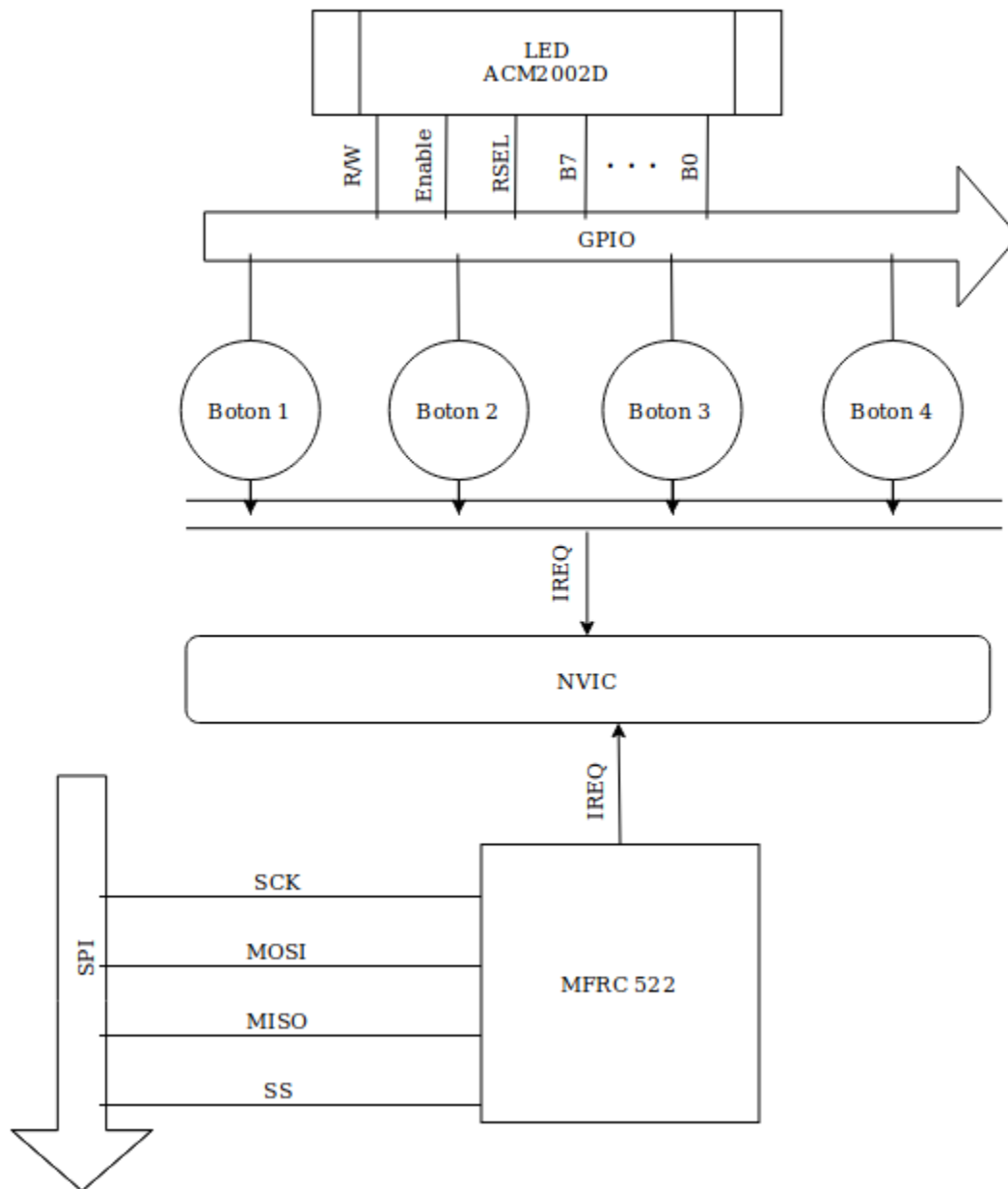


Conexiones necesarias entre MFRC522 y Procesador:

- SCK conectado en P3\_3 configurado como función SSP0\_SCK.
- MOSI conectado en P3\_7 configurado como función SSP0\_MOSI.
- MISO conectado en P3\_8 configurado como función SSP0\_MSIO.
- SS conectado en P3\_6 configurado como función SSP0\_SSEL.
- Se utilizará también una línea de interrupción.

## V. Esquema general de Hardware

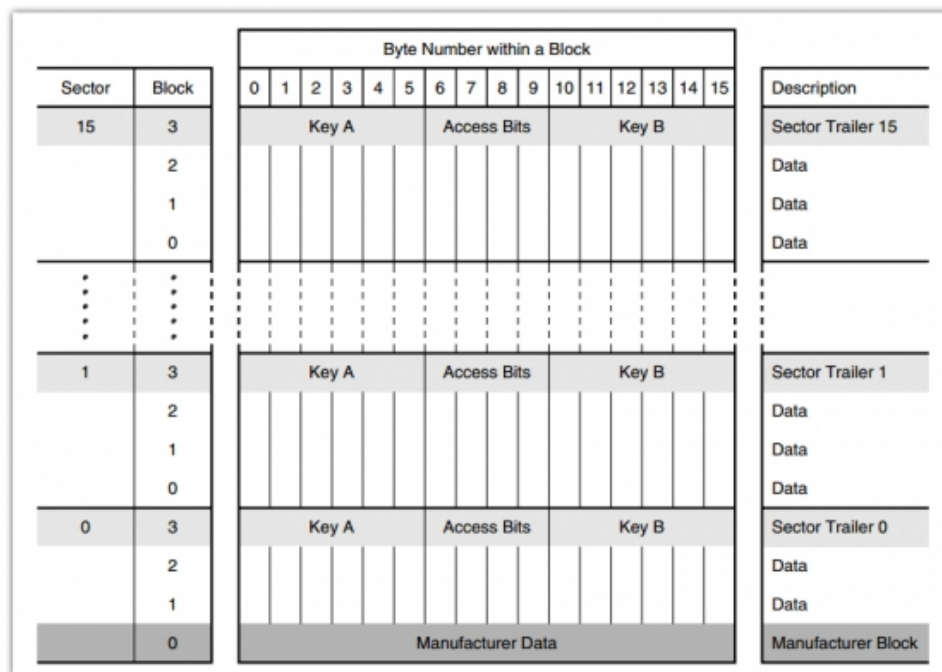
El esquema general de hardware completo está compuesto de la siguiente manera:



## VI. Tarjeta MIFARE

La tarjeta que vamos a implementar y que utilizaran los usuarios son las tarjetas MIFARE: una tecnología de tarjetas inteligentes sin contacto (TISC), equivalente a las 3 primeras partes de la norma ISO 14443 Tipo A de 13.56 MHz con protocolo de alto nivel, con una distancia típica de lectura de 10 cm. La misma cuenta con zonas de memoria que permiten almacenar información y que tiene un cierto nivel de seguridad para acceder a dicha información (encriptación)

### Almacenamiento de la información




La información de la tarjeta es almacenada en una memoria de tipo EEPROM de 1K. La misma 16 sectores, que a su vez se subdividen en 4 bloques por sector de 16 bytes cada uno, lo que permite que una única tarjeta se pueda utilizar para diferentes servicios.

El acceso a cada uno de los diferentes bloques es definido por el usuario. Esto significa que cada bloque puede ser:

- público (no necesita clave y puede ser leído por cualquier aplicación)
- privado (necesita clave para poder leer / escribir la información creando zonas seguras).

Esta característica hace que la tarjeta MIFARE sea suficientemente segura para uso como tarjeta monedero.



Los sectores utilizan dos claves de acceso llamadas 'A' y 'B'. Estas llaves se almacenan en el cuarto bloque junto con los permisos de acceso a cada uno de los tres bloques. Estos permisos pueden ser: lectura, escritura, descuento o incremento (para bloques de valor).

Existe un bloque especial que está localizado en el bloque de datos 0 del sector 0, que alberga la siguiente información en modo solo lectura:

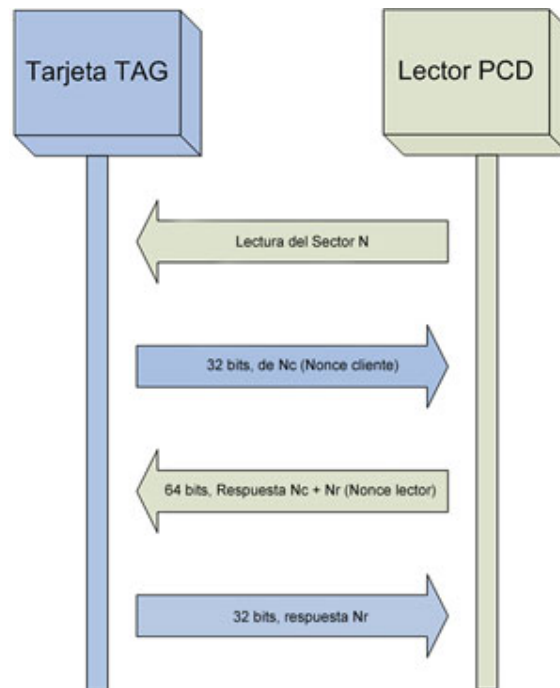
- UID – Identificador único de la tarjeta.
- BCC – bit count check, calculado por la sucesivas XORs de los bytes del UID.
- 11 bytes de datos que identifican al fabricante del tag.

## Comunicación cifrada

Una vez que se acerca la tarjeta a un lector, ésta se activa e inicia un proceso de intercambio con el lector para establecer una comunicación cifrada. Este proceso es igual con todas las tarjetas y está diseñado para proveer protección contra escucha del canal, y no para autenticar la tarjeta o el lector.

Después de establecer un canal cifrado, la tarjeta envía un código de identificación de conexión, que usualmente es el número de serie de la tarjeta, aunque la norma ISO 14443 dice que este número puede ser aleatorio. Con este número de conexión el lector está en capacidad de realizar cualquier operación en la tarjeta.

## Proceso de autenticación



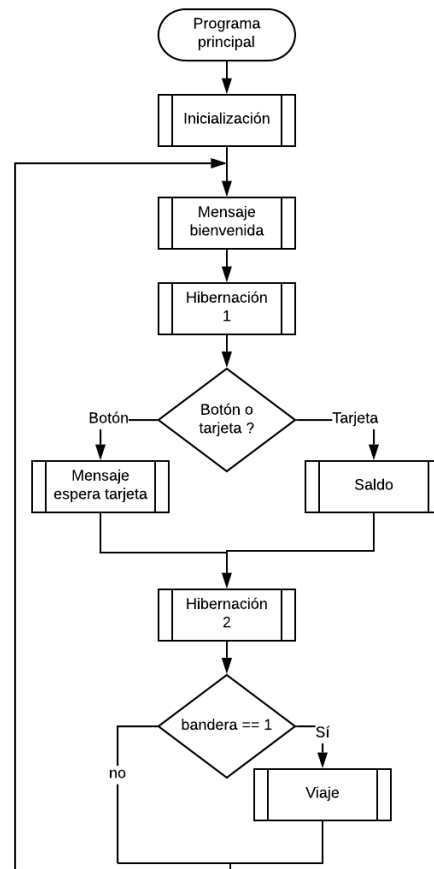
1. En primera instancia el lector le comunica a la tarjeta que quiere realizar una operación sobre un sector de datos determinado.
2. La tarjeta en ese momento remite un número aleatorio de 32 bits a modo de desafío, para que sea cifrado con la clave privada compartida previamente.
3. Como respuesta, el lector remite el desafío cifrado y un número aleatorio para que el tag lo cifre con la clave privada, generando una trama de 64 bits.
4. En última instancia la tarjeta le envía al lector su desafío cifrado.
5. En este momento ambos tienen la certeza de que los dispositivos son legítimos.

## VII. Estructura de datos

Para la implementación del software de nuestro dispositivo, utilizaremos las siguientes variables:

Variable	Descripción	Tamaño
<b>BOTÓN</b>	Detección de botón.	1 Byte
<b>SALDO</b>	Contiene el saldo de la tarjeta vinculada.	10 Bytes
<b>COSTO1</b>	Tarifa correspondiente al viaje 1.	1 Byte
<b>COSTO2</b>	Tarifa correspondiente al viaje 2.	1 Byte
<b>COSTO3</b>	Tarifa correspondiente al viaje 3.	1 Byte
<b>COSTO4</b>	Tarifa correspondiente al viaje 4.	1 Byte
<b>LED_A</b>	Enciende LED amarillo.	1 Byte
<b>LED_R</b>	Enciende LED rojo.	1 Byte
<b>LED_V</b>	Enciende LED verde.	1 Byte
<b>BANDERA</b>	Detecta proximidad de una tarjeta.	1 Byte
<b>DISPLAY_OUT</b>	Salida del display LCD.	1 Byte
<b>MODOTARJETA</b>	Indica si la tarjeta debe decrementar o no saldo	1 Byte
<b>MODOLUZ</b>	Indica que LEDs debe encender	1 Byte

## VIII. Diagramas de flujo

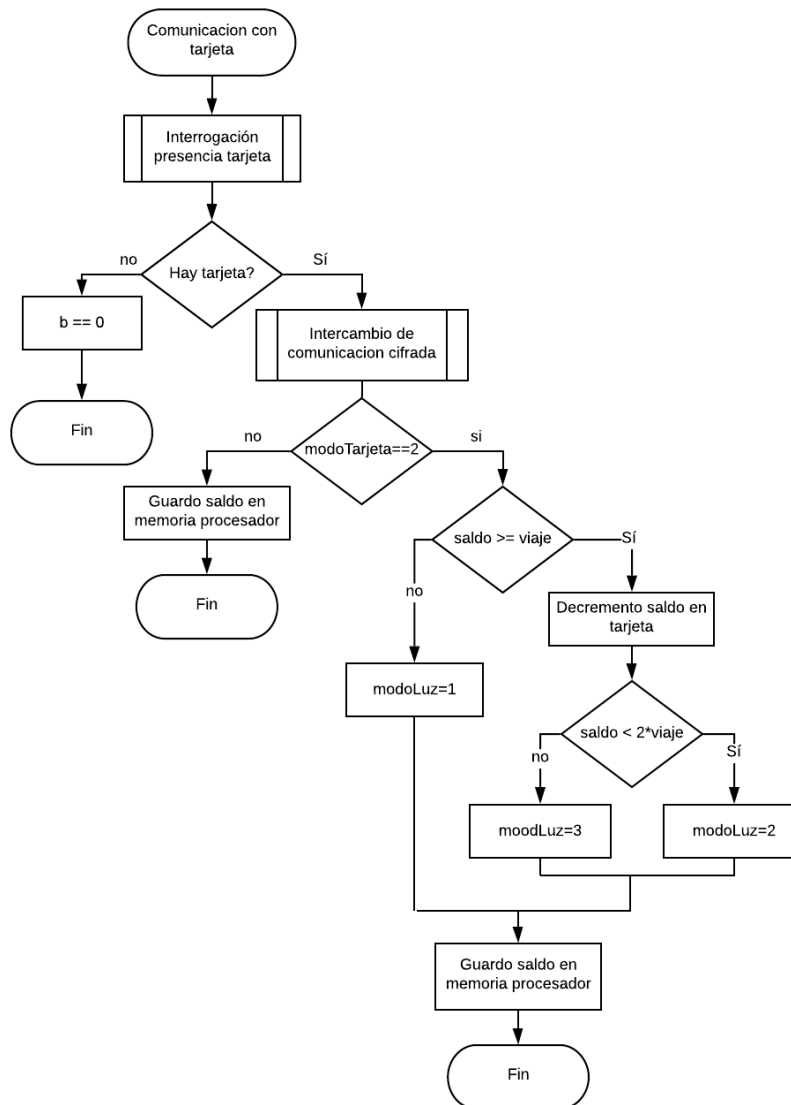


- Cuando se enciende el dispositivo, realiza la inicialización correspondiente (se configuran los puertos, inicializa las variables necesarias, etc)
- Muestra el mensaje de bienvenida por pantalla, a la espera de una tarjeta o botón.
- Entra en un estado de hibernación, para que el consumo de energía sea mínimo. Saldrá de esta rutina cuando reciba una interrupción, ya sea por presencia de una tarjeta o por un botón:
  - Si se acercó una tarjeta, ingresa a la rutina de Saldo y muestro por pantalla el saldo actual y está a la espera de que presione un botón para elegir viaje
  - Si se presionó un botón, el dispositivo está a la espera de una tarjeta
- Luego, el dispositivo entra en un segundo modo de hibernación similar al anterior: espera la siguiente interrupción correspondiente (tarjeta o botón) pero tiene un timer de 10 segundos. Sí el mismo llega a 0, sale de la rutina con una bandera = 0, no abonará ningún viaje y volverá a mostrar el mensaje de bienvenida.



- Cuando se realiza la consulta de saldo, la variable **modoTarjeta** se pone en 1, para que la rutina de comunicación con la tarjeta sepa que es solo para consulta, y no un decremento de saldo.
- Una vez realizada la comunicación, la rutina retorna con el saldo actual, y lo muestra por pantalla





El sistema está a la espera de que una tarjeta se acerque al lector:

- Si no la encuentra, sale de la subrutina con **bandera = 0**
- Si la encuentra, realiza el Intercambio de comunicación cifrada: Realiza el proceso de autenticación y establece el canal de comunicación cifrada. Una vez realizada, se pueden hacer las operaciones necesarias.

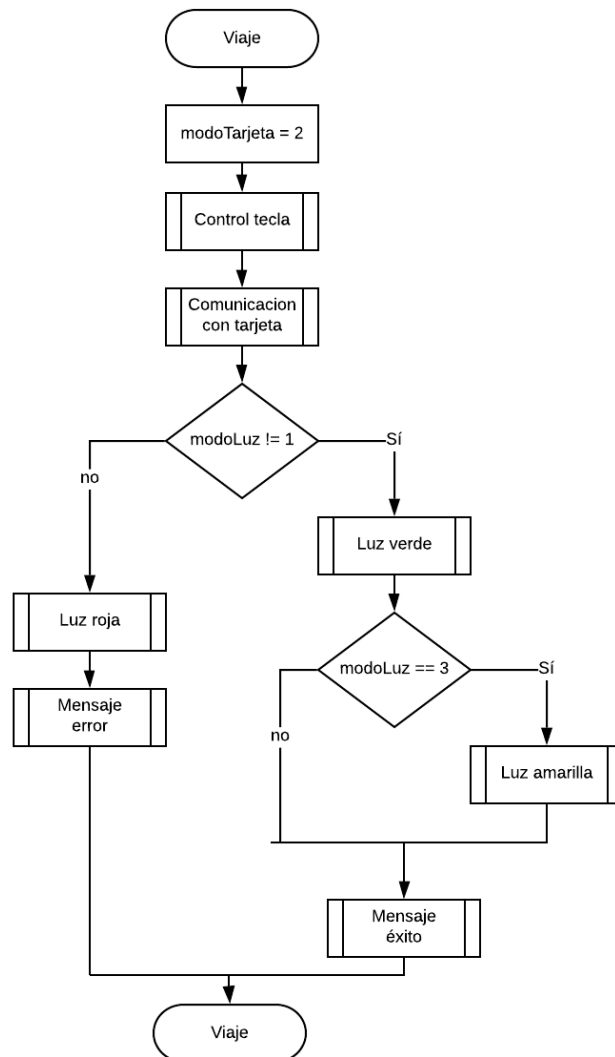
En este paso, obtiene el saldo actual y la conexión sigue establecida para hacer el decremento (sí es posible).



Luego, controla qué operación quiere hacer con la tarjeta:

- Sí el ***modoTarjeta* != 2**, significa que sólo es consulta de saldo, por lo que lee el saldo en tarjeta, guarda su valor en memoria y finaliza la comunicación.
- Sí el ***modoTarjeta* == 2**, debe controlar si puede cobrar el viaje.
  - Si el saldo es suficiente, decrementa el saldo de la tarjeta, y controla si debe encender sólo la luz verde, o junto con la amarilla
  - Si el saldo es insuficiente, enciende la luz roja

En cualquier caso que se presente, debe guardar el saldo de la tarjeta (ya sea el original, o el decrementado) para poder mostrarlo por pantalla



- La rutina pone el **modoTarjeta** en 2, para que sepa que deberá realizar un decremento de saldo (de ser posible).
- Realiza el control de qué tecla se presiono, para poder asignar el valor del viaje correspondiente
- Establece la comunicación con la tarjeta y realiza (o no) el pago. La rutina retornará una variable **modoLuz** que puede tomar 3 valores:
  - **modoLuz=1**, el pago no pudo realizarse por saldo insuficiente
  - **modoLuz=2**, el pago se realizó correctamente
  - **modoLuz=3**, el pago pudo realizarse, pero el saldo restante es menor a 2 viajes

## IX. Mejoras futuras

Proyectandonos a futuro, proponemos las siguientes mejoras:

- Implementación de una interfaz alternativa inclusiva para no videntes. Esta contará con un parlante y botones con su respectivo significado en código Braille.
- Implementación de rastreo satelital y botón de pánico en caso de siniestro.
- Molinete para controlar el flujo de personas antes de ingresar a la unidad, asegurandonos:
  - Total independencia del chofer, abstrayéndose de controlar el pago del boleto y la interacción con el usuario.
  - Identificar al causante del siniestro, en caso de haber uno.
  - Establecer un límite máximo de pasajeros a bordo