

Diseño WeightLogger

RFID control

Este documento detalla el diseño para el sistema weighlogger, incluyendo electrónica y Software.

Cordoba, Argentina, 2018-08-20
ELCAN SOFTWARE



Mario O. Villarroel
System Design/Developer
movilla@elcansoftware.com

Pablo Giachero
Product Owner
pgiachero@elcansoftware.com

1. Control de version

Version	Author	Reviewer	Approver	Date	Change log
1.0	Mario O. Villarroel			2018-08-20	Versión inicial
2.0	Mario O. Villarroel			2018-08-28	Traducción al español
2.1	Mario O. Villarroel			2018-09-04	Detalles sistema administrativo

2. Lista de distribución del proyecto

Name	Role	E-mail	Date	Signature
Mario O. Villarroel	System Engineer	movilla@elcansoftware.com	2018-08-20	-
Pablo Giachero	Product Owner	pgiachero@elcansoftware.com	2018-08-20	-
Cliente Beta	Beta tester	cliente@elcansoftware.com	2018-08-20	-

3. Propósito del documento

Este documento detallará el sistema para registrar el peso de un camion e identificar el conductor, para mantener un control de la carga transportada y los tiempos de ejecución.



4. Requerimientos

Los siguientes requerimientos han sido obtenidos por el PO¹ en reuniones con el cliente final.

1. Debe ser capaz de funcionar con una red inalámbrica.
2. Debe usar sólo un cable, para su alimentación.
3. No debe requerir soporte especial para su montaje.
4. Debe haber un dispositivo de control por cada báscula.
5. Hay más de una báscula en la ubicación del cliente.
6. Debe tener un software que organiza y ayuda con las tareas de control.
7. Debe usar la tecnología RFID de amplia disponibilidad en el mercado.
8. debe contar con un display LED y un buzzer para sonido.
9. Podría usar energía solar como fuente de alimentación si el costo del proyecto lo permite.
10. Los dispositivos permanecerán en el exterior, necesitando cuidados mínimos para su funcionamiento.
11. Habrá un servidor conectado a la red donde los dispositivos enviarán la información “directamente”.
12. El servidor estará conectado y disponible con un uptime mayor o igual al 99 % del tiempo de trabajo del cliente final.
13. El software administrativo será basado en web, permitiendo el acceso desde múltiples dispositivos.
14. Los datos del sistema se mantendrán en la locación del cliente, no siendo necesario el uso de internet para su utilización.
15. Los datos podrán ser accedidos desde Internet con un set especial de credenciales, usando estándares seguros (https, claves cifradas, etc.)
16. El diseño debera considerar que cada dispositivo electrónico sea capaz de trabajar por su cuenta.
17. Los dispositivos no almacenarán internamente los datos de conductor y peso, solo los retransmitirán por la red al servidor.
18. El costo total de cada sistema completo deberá permanecer por debajo de AR\$10000, esto incluye dos dispositivos electrónicos y el sistema de administración.

TODO: Agregar mas requerimientos acerca del sistema administrativo.

5. Diseño

5.1. Electrónica

Para la electrónica hemos seleccionado la plataforma AVR, de amplia difusión y facilidad de adquisición en el mercado local.

Este chipset también permite el uso del lenguaje C++ para el código fuente, lo que permite una mantenibilidad mejorada y duración en el tiempo del proyecto debido a esto.

El dispositivo final usará una red inalámbrica standard 802.11 para enviar datos, como también el protocolo TCP/IP para transportarlos.

El dispositivo se colocará en un gabinete con estetica adecuada y resistencia al ambiente de trabajo.

Para la configuración del dispositivo se contará con un pulsador en la placa, apretando ese boton se reconfigura la red wifi del mismo mediante WPS, al concluir la configuración se mostrará el IP obtenido en el display del artefacto.

El IP configurado en el dispositivo se mostrará cada vez que se inicie el mismo.

5.2. Casos de uso de la electrónica

En la figura 1, pueden verse los casos de uso del equipo electrónico a diseñar.

¹Product Owner

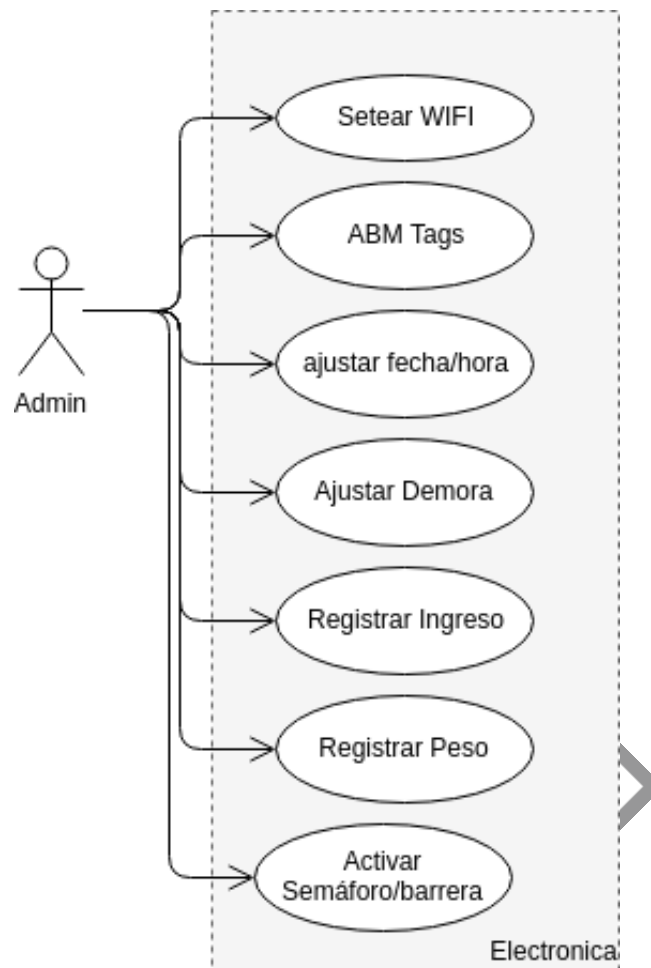


Figura 1: Casos de uso Electrónica

5.3. Secuencia para un registro

Para registrar un evento se seguirá la secuencia demostrada en la figura 5.3

5.4. Secuencia para agregar un TAG

La secuencia en la figura 3 muestra los pasos requeridos para agregar un tag como conocido en la base del equipo, los tags que no estén como conocidos causarían una alerta en el sistema y no cambiará el semáforo o activará la barrera.

5.4.1. Diseño del PCB

El diseño de la placa para el sistema electrónico se detalla en la figura 4

Al no disponer de más de una interface serial, la conexión con el wifi y/o balanza (a determinar en pruebas) se hará mediante I2C, con la placa mostrada en la figura 5

5.5. Protocolo de comando

El sistema electrónico entenderá un protocolo de comandos mínimo que le permita ejercer todas sus funciones, para la versión 1.0 del mismo se ha diseñado el siguiente:

- TIME-ADJUST: Para permitir el ajuste del rtc incorporado en el equipo.
- STORE-CARD: Almacenar tarjeta nueva en posición determinada.

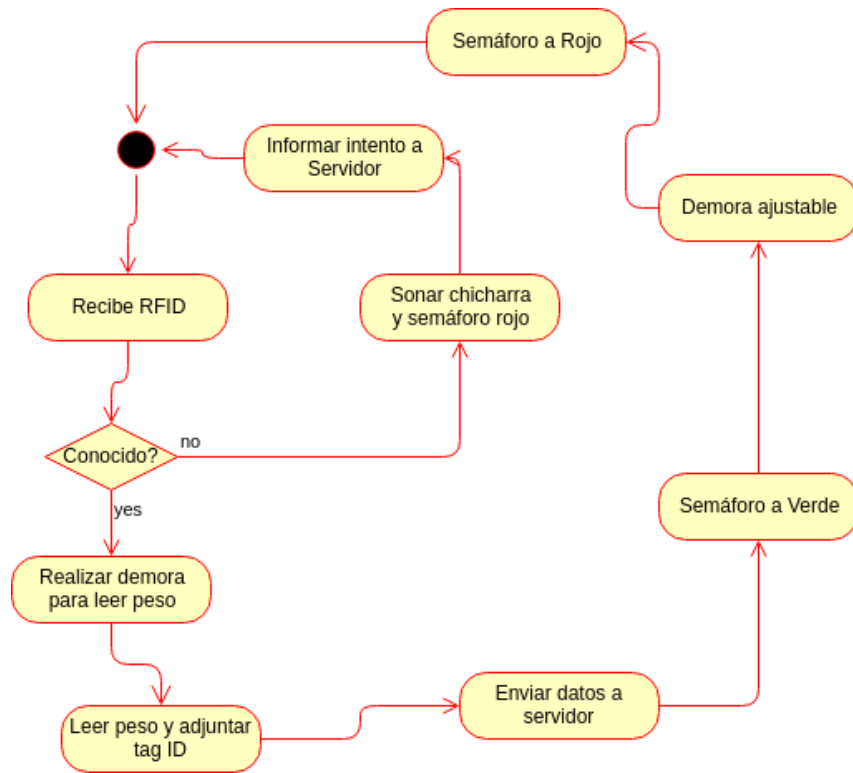


Figura 2: Diagrama de secuencia

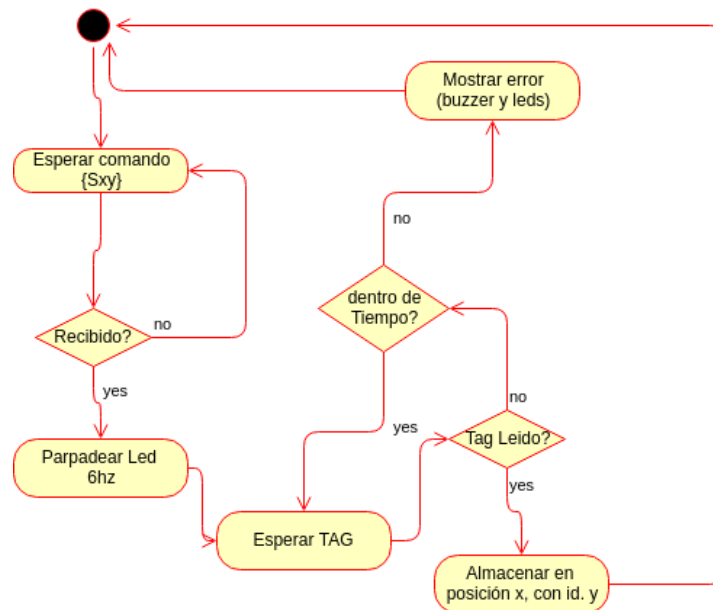


Figura 3: Secuencia registro de tag

- DUMP-EEPROM: Descargar el contenido de la eeprom interna con los tags conocidos.
- ERASE-CARD: Borrar un tag conocido de la memoria EEPROM.
- REJECTED: Envío de un intento de ingreso rechazado.
- RECORD: Envío de un registro completo con peso, fecha/hora y TAG ID.

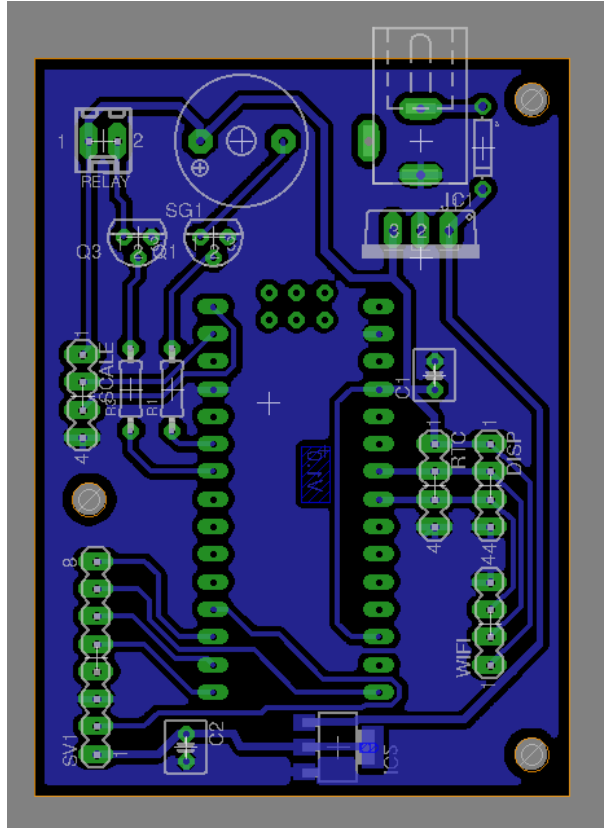


Figura 4: PCB v1.0

Este protocolo mínimo será soportado por la versión 1.0 y se recibirá mediante protocolo TCP/IP por la red inalámbrica. El detalle de los bytes enviados y respuesta se detallará a continuación.

TIME-ADJUST [Twxyz] el comando comienza con la letra T mayúscula acompañado de 4 bytes que representan el tiempo en formato unix timestamp, comenzando por el LSB hacia el MSB.

STORE-CARD [Sxy] Comando de letra S mayúscula acompañado de dos bytes, x para posición de 0 a 200, y para identificador de tag que será exportado al enviar registro. El ID del tag a guardar será leído por el dispositivo.

DUMP-EEPROM [D0] Comando de letra D mayúscula, no requiere parámetro adicional.

ERASE-CARD [Ex] Comando de letra E mayúscula, acompañado de byte x con la posición a borrar, de 0 a 200.

REJECTED [Rwxyzabcd] Comando con letra R mayúscula, acompañado de los 4 bytes de unixtimestamp como en TA² y 4 bytes del TAG ID rechazado.

RECORD [Awxyzabcdefgi] Comando con la letra A mayúscula, acompañado de los 4 bytes de unixtimestamp igual que en TA, los siete bytes del peso leídos de la báscula y el ID del tag leído en el último byte.

5.6. Mensajes del sistema

El dispositivo electrónico contará con mensajes que mostrará a travez de su display, los mismos se detallan a continuación:

- En modo espera mostrará “Esperando” y la hora y fecha actual.
- Al recibir un tag conocido mostrará: “Acceso permitido”.

²TIME-ADJUST

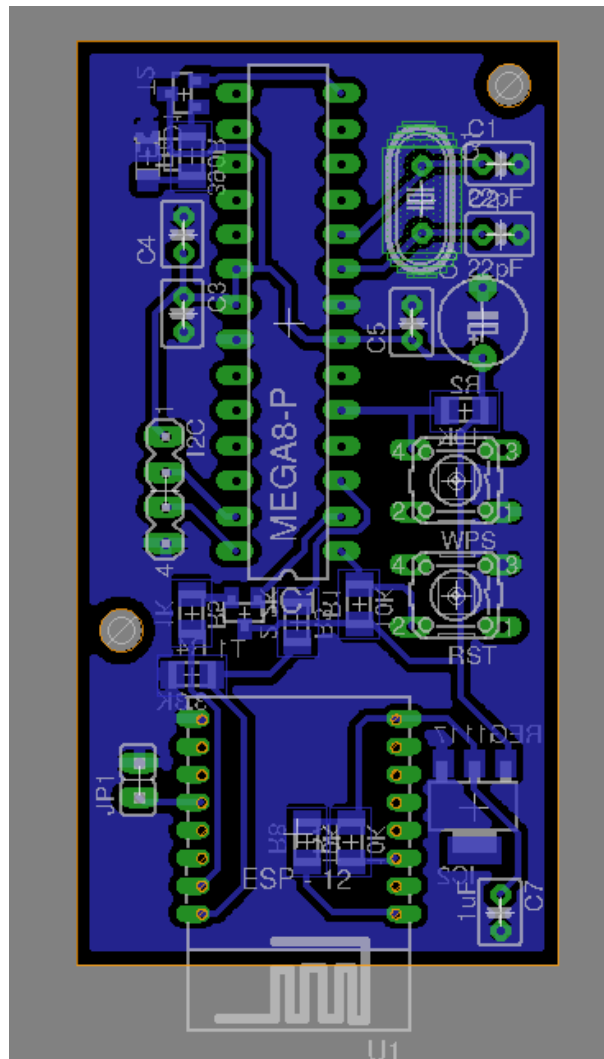


Figura 5: PCB interface I2C-Serial

- Cuando lea un tag desconocido mostrará: “Acceso negado, informando”.
- Al confirmar pesaje o luego de la demora mostrará “Avance...”.
- En Configuración mostrará el IP obtenido por el wifi.
- En caso de error mostrará “Error” y un código detallado en la sección 5.6.1

5.6.1. Códigos de error

Los siguientes códigos de error se manejarán en el dispositivo:

Código	Descripción
501	Falla RTC
502	Falla RFID
503	Falla WiFi
504	Falla EEPROM

5.7. Software de Administración

El software de administración requerido será desarrollado en Ruby On Rails para mejor velocidad de desarrollo.

5.7.1. Casos de uso para el sistema

Los casos de uso que se han considerado para el sistema administrativo se detallan en la figura 6, se incorporarán más casos en las subsiguientes iteraciones del sistema.

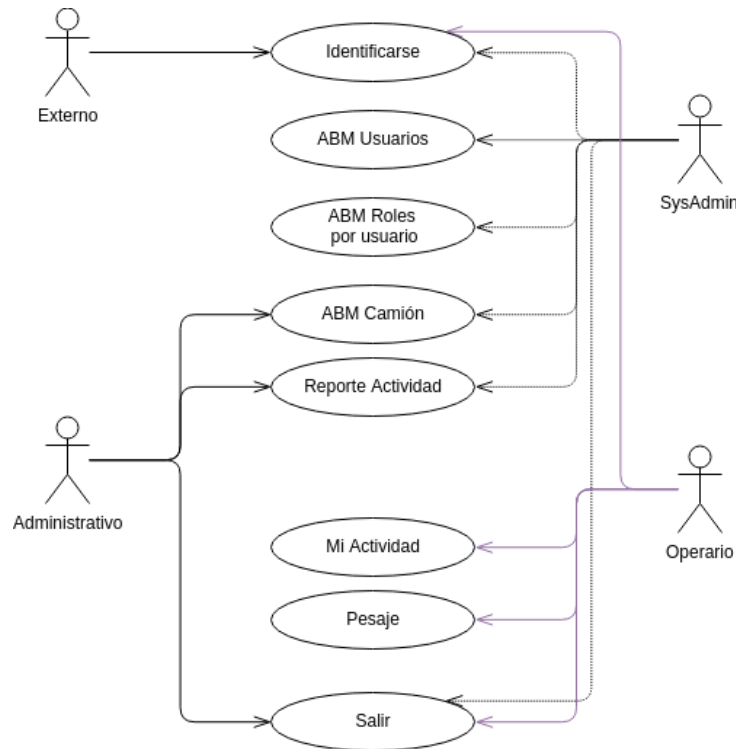


Figura 6: Casos de uso sistema Administrativo

5.7.2. Diagrama de entidad relación

Las entidades y sus relaciones se detallan en la figura 7, se evolucionarán sus partes acorde se avance en las versiones del sistema desarrollado.

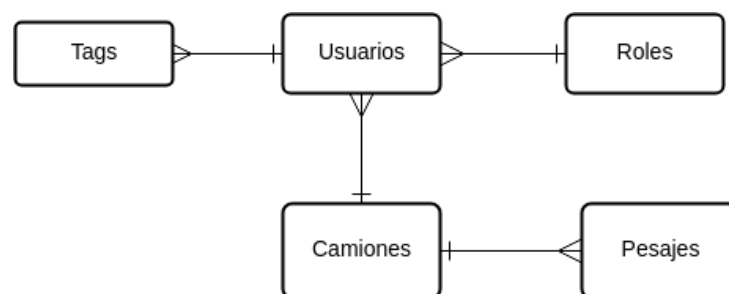


Figura 7: Diagrama Entidad-Relación

5.7.3. Desiciones de diseño del sistema admin

Se utilizará la mayor cantidad de gemas disponibles para el sistema, de forma tal que ascelere el desarrollo y se pueda entregar rápidamente algo de utilidad al cliente final.

Utilizaremos al menos las siguientes gemas:

- Devise, para manejo de autenticación.
- ActiveAdmin, para ABMs rápidos.
- etc.