**Calendario Sprint 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Investigación funcionalidad balanza bobinas | Investigar el datapath de la balanza yendo a hablar con el fabricante | 7 | Sin realizar | 7/01/2016 |  |
| Realizar tests | Realizar los tests respectivos a la aplicación C# y android | 5 | Sin realizar | 7/01/2016 |  |
| Investigación lector QR | Leeríamos el código del lector de google para poder informarnos bien e implementarlo en nuestra aplicación | 5 | Sin realizar | 8/02/2016 | Sprint3 |

Issues sprint3:

Hitos Sprint3:

**Funcionalidades balanza de bobina**



La balanza provista por el cliente, es una balanza electrónica con la capacidad de pesar bobinas de papel de hasta 1000kg. El modelo de la misma es UMC 555 GEA y fue vendida por CAIR SRL.

Una vez estudiado la hoja de datos del modelo de la balanza, se visito al fabricante de dicha balanza, para confirmar las configuraciones apropiadas.

En la Ilustracion1 podemos observar la hoja de configuración entregada por el fabricante de balanzas electrónicas CAIR SRL, en donde allí detalla las configuraciones que deben aplicarse para la salida en puerto COM.

Ilustración

Configuración:

- 9600 bautios

- 7 bits de paridad impar

Conexiones Borneras:

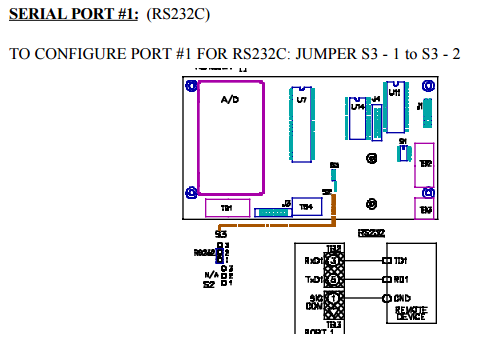
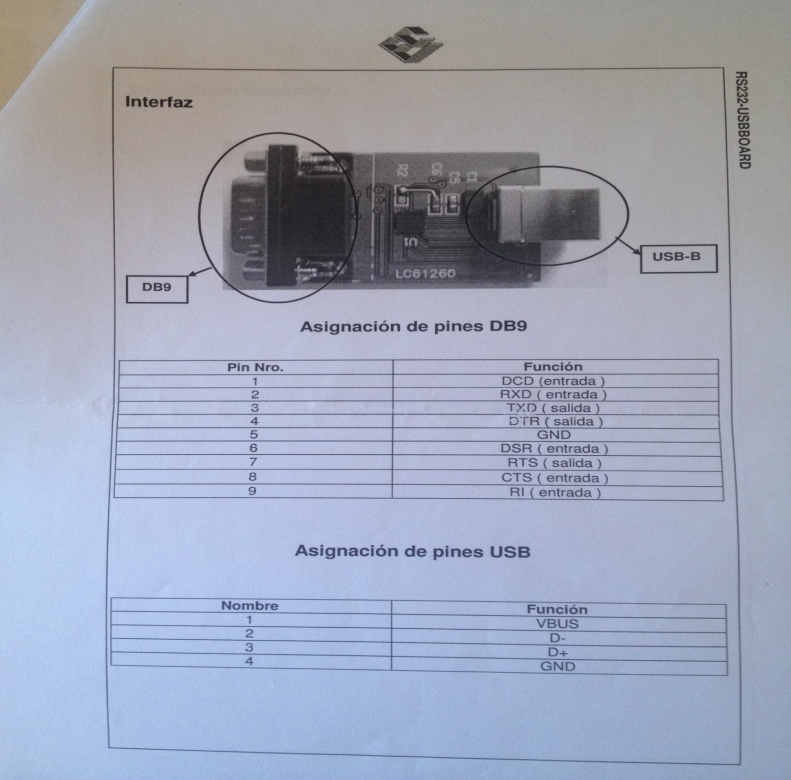
Como se puede observar en la ilustración 2, se ve a grandes razgos el circuito del cabezal digital que toma el peso de la balanza. Allí se indica cuales son las borneras de Tx, Rx y GND para enviar por RS232 (DB9) la salida del peso de la misma.

Ilustración   
Seccion 6 (Serial OutPut)

Para llevarlo a la PC, lo que se hizo fue comprar un adaptador de DB9 a usb.

A nivel programación lo que se debió hacer es incluir una librería para C#, System.IO.Ports y poder recibir el dato transmitido en el puerto COM conectado.

Adaptador DB9-USB



Ilustración

En la ilustración 3 se pueden observar cual fue el adaptador adquirido y cuales son los pines del Tx,Rx y GND para conectar a las borneras de la balanza.

**Tests/Code coverage**

**Código QR**

****

Para la leer codigos QR nos basamos en un proyecto open source llamado Zxing cuyo repositorio es el siguiente:

<https://github.com/zxing/zxing>

ZXing ("paso de cebra") es un multi-formato de biblioteca de procesamiento de imágenes de código de barras de código abierto 1D / 2D implementado en Java, con los puertos a otros idiomas.

**Formatos Soportados**

|  |  |  |
| --- | --- | --- |
| **1D product** | **1D industrial** | **2D** |
|  |  |  |
|  |
| UPC-A | Code 39 | QR Code |
| UPC-E | Code 93 | Data Matrix |
| EAN-8 | Code 128 | Aztec (beta) |
| EAN-13 | Codabar | PDF 417 (beta) |
|  | ITF |  |
|  | RSS-14 |  |
|  | RSS-Expanded |  |