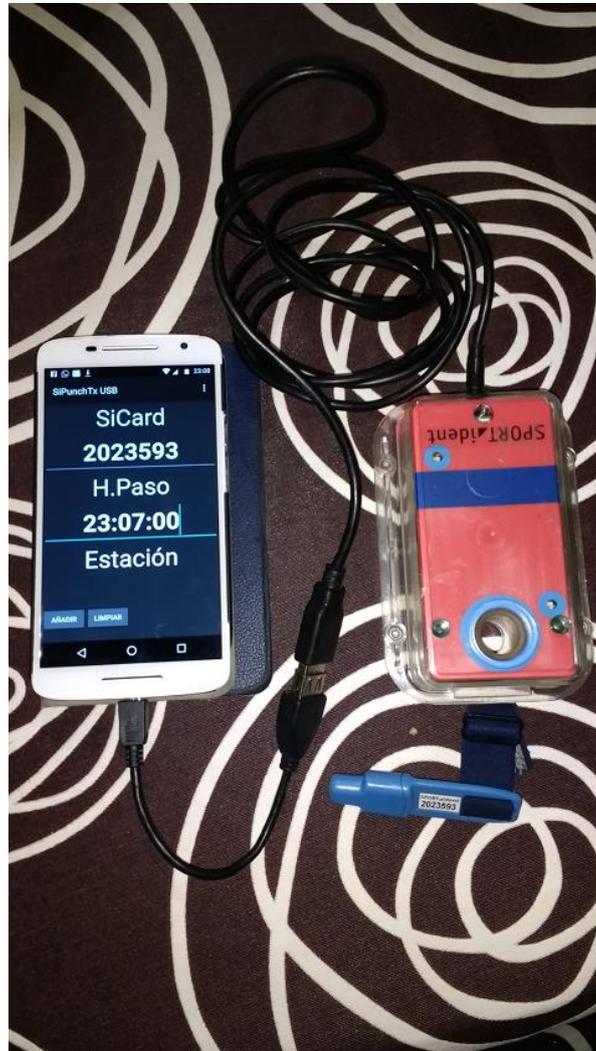


Intermediate Control Points the easy way

OE2010 Integration



Intro

There is a system to control intermediate points on an orienteering race, but it relies on a complete DBMS managed by a competed user. The idea of this new version is to take advantage of the functionalities provided by the software OE2010. This solution only works with the sportident system.

OE2010 contains a commentator module. Several functions are related to intermediate control points timing.

OE2010 can operate with a unique computer or in a network. The software can connect to stations intended to work as intermediate control points. This is done through the use of a client form. Normally, PCs on a network physically connected to stations launch a client form inside OE2010 and receive in real time information about punches. Other PC (or just one performing all of the actions) launches a server form that is in charge of communicating with clients and save intermediate timing info into the database.

Another way of communication is through a web client form. The idea is that the punches from the stations are saved in a web server. The web client form connects to the web server, reads the information and put it available to the server form.

This easy way is the solution provided in this document. At these days, OE2010 interacts only with one web server, called “ROC olresultat”. Basically, it’s a ASP page in a server that is called from OE2010 with some specific parameters, and gives an answer in another specific format.

The implementation of the solution consists on replacing the ROC server by another that “speaks” the same language (format of the request, format of the answer)

Description of the system

List of elements that are needed (as seen on the top figure):

- Readout station, acting as a normal control or finish. “send data” property should be checked. This station can be USB, RS232 or BSF8-SRR. The station should be configured with these parameters: **extended protocol, autosend**. The baudrate can be configured to **4800** or **38400** as the app includes a functionality to set this value on the serial port
- If it’s a RS232, Sportident RS232 to USB converter is needed
- Female USB to male microUSB, OTG Cable
- Android Smartphone, USB OTG capable
- SIM Card with Internet connectivity for sending data (or Wifi, if exists at the location of the intermediate control)
- APK SiPunchTxUSB Min (connects to station, reads punches and sends data to server)
- OE2010 with commentator capabilities

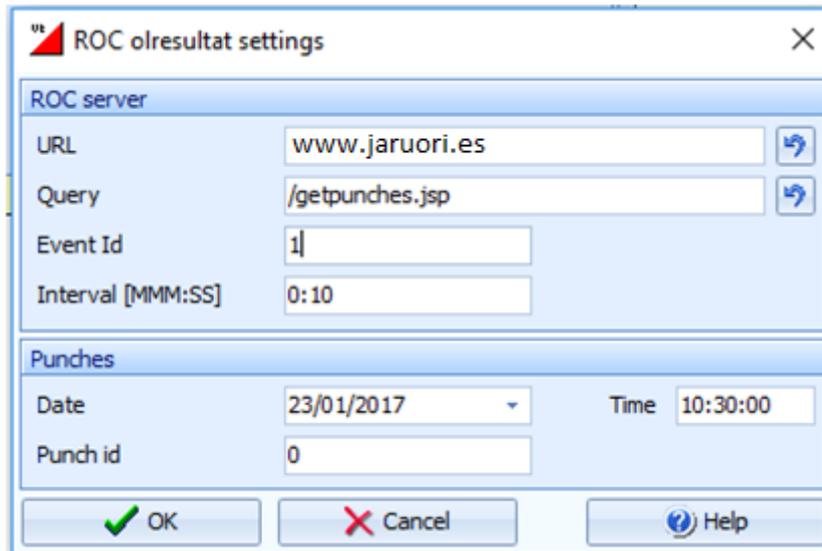
Important notice

This is not a product. It’s just the result of a benchwork where the author implements and tests ideas and solutions related to the sport of orienteering and IT. So, there are no fancy designs, complete reference guides or user’s support. The current status of the system is beta test and is planned to be used in a real competition next February, 2017.

OE2010 configuration

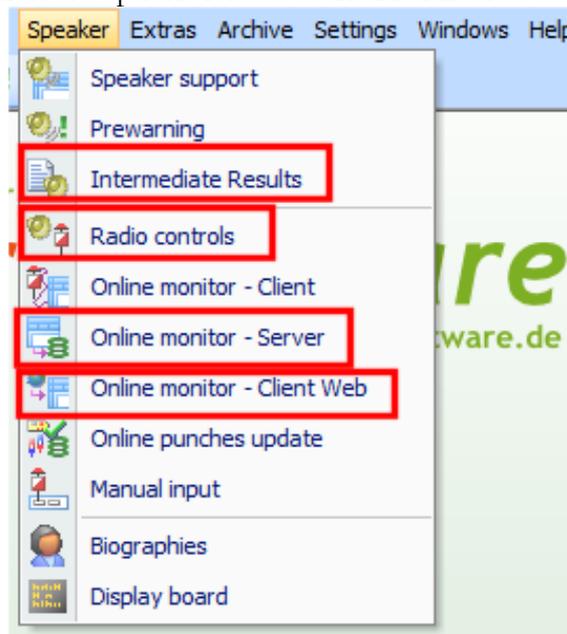
OE2010 has a web client form that lets the user configure some important parameters that are in charge of performing a correct communication with the web server (the one that stores the punches on internet)

- **URL:** in origin is: roc.olresultat.se. In our case it will be: **www.jaruori.es**
- **Query:** in origin is: /getpunches.asp. It will be: **/getpunches.jsp**
- **Event ID (unitId):** a number. The concept of event will be mandatory just for this reason
- **Update interval:** used by OE2010 to know how many time has to pass before requery
- **Date:** it allows to limit the number of records for reading (starting from that date)
- **Time:** it allows to limit the number of records for reading (starting from that time)
- **Last reading (lastId):** an identifier for punches. It allows starting from a given number instead of requerying all.



The web client form relies on the “Online monitor – Server” form, because the web client form transmits the data read from the web server to the online monitor server, and this is in charge of saving the information about intermediate controls to the OE2010 database.

The next image shows the main menu options related to the intermediate controls functionality.

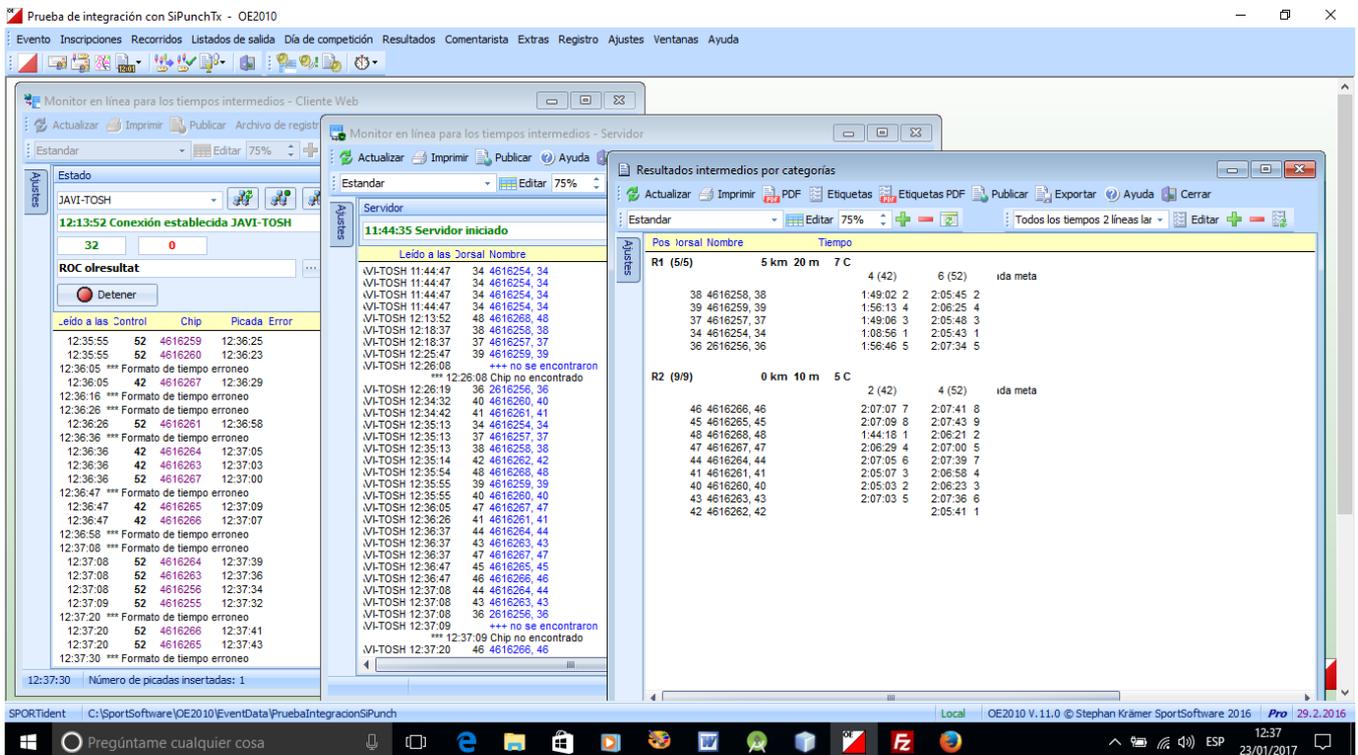


First of all, the user must define the identifiers of the intermediate controls.



The "Finish punch" is mandatory. If the user wants to send information from the finish line, he only has to define a finish station with an identifier lower than 31. For example, if the finish station has a value of 5, when OE2010 receives a punch from that station it decides that it's a value from the finish line.

The next image shows 3 forms in 1. The example shows that there is one computer acting as the server, as a web client (requesting and receiving data from the web server) and as a client that shows intermediate results to the speaker (or the public)



The form in the left is the web client. It's connected to the web server and to the OE server. It's requesting data to the web server periodically and transmitting them to the OE server.

The form in the middle is the OE server. It's connected and receiving data from the clients (in this case, only from the web client) and then saving the information to the database.

The form in the right shows the intermediate results.

User's Web Side

From a user point of view there is only one need. That is the need of an event, only because OE2010 uses that concept in order to filter the queries (avoiding the possibility of confusion between events organized by different clubs or federations). Web side pages are located here:

<http://www.jaruori.es/minintro.jsp>

So, there are a few functionalities on the web server in order to manage events. One user (no need for user registration) can create an event providing some data (description, date). There is no need for registration but a user and password should be provided in order to limit the number of persons that can edit that basic information and the number of devices that can send punches to the web server.

The user and password is linked to an event. Using that data, the user can edit the basic information of the event through the use of other web pages.

User's Smartphone Side

Only Android is supported. The easy way requires the use of a smartphone that supports USB-OTG. It exists a similar app that can use Bluetooth, but needs extra hardware (very easy to build but it will be published in the future)

A readout station is connected to the smartphone through a USB-OTG cable. The smartphone runs the app. It can be downloaded here:

<http://www.jaruori.es/cpiminimo/SiPunchTxUsb.apk> -> **Android <8**

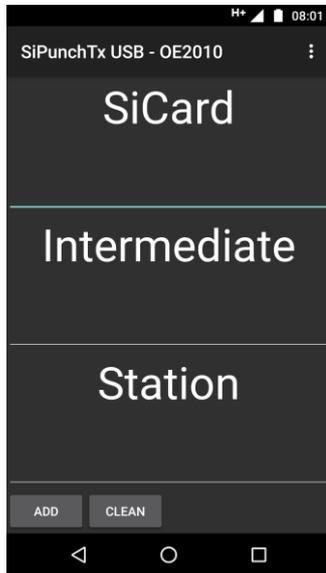
<http://www.jaruori.es/cpiminimo/SiPunchTxUsb3.apk> -> **Android >=8 (Oreo)**

The first thing to do is logging into the system in order to assign the station to a given event. From that moment, every punch is firstly queued into the smartphone until it can be transmitted to the web server.

Connection between smartphone and readout station is not mandatory. A user can act as a controller standing near a control and writing directly several punching values. In order to ease the task, the app allows the user to change between SiCard value and Bib number. This can be used only if the event has a link between those values. This can be done in the web side of the app, uploading a CSV file containing pairs of Bib numbers and SiCards separated by semicolon symbols. The first row should contain labels for those fields (Bib;SiCard)

Appendix. Snapshots from the Smartphone Side

The app has only a few screens and it's very simple to use. The main screen shows the data read from the station or manually written by the user. The normal use of the app is unattended but a user can manage it in order to send data by hand. After connecting a readout station to the device, the app can be launched.

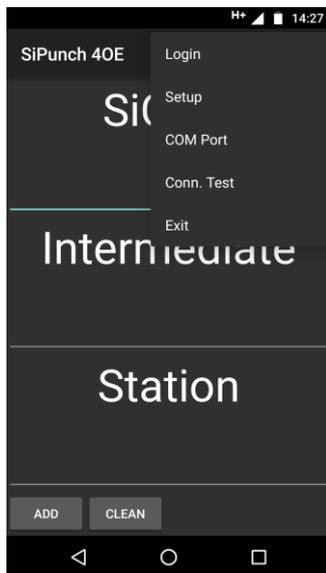


SiCard is the number of the Sportident card associated to a runner. Remember that only sportident hardware is supported.

Intermediate (time) is the time of the punch. The format is HH:mm:ss (10:40:02, 17:21:30, etc)

Station is the code of the sportident station (31, 32, etc). A number less than 31 is treated by OE2010 as a finish station.

The app works in background so the user can change to another app, for example to make a call or check the last whatsapp messages. Anyway, the best thing is to launch the app, lock the screen and let it work without intervention of the user.



The main menu has only 5 options:

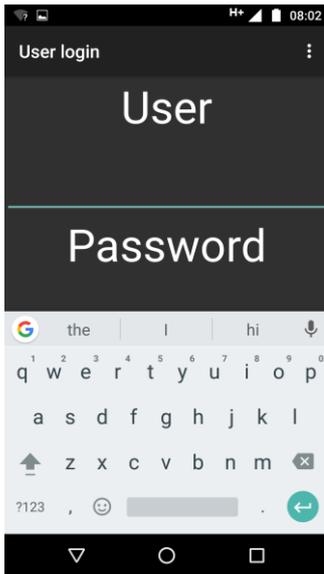
Login shows a screen that allows to log into an event.

Setup is intended to configure some parameters

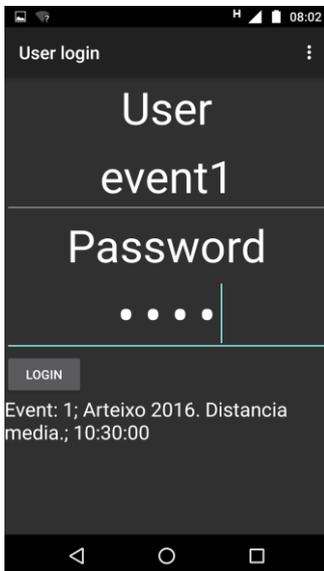
COM Port allows to set the parameters of the serial port

Conn. Test allows to test whether the device can connect to the web server

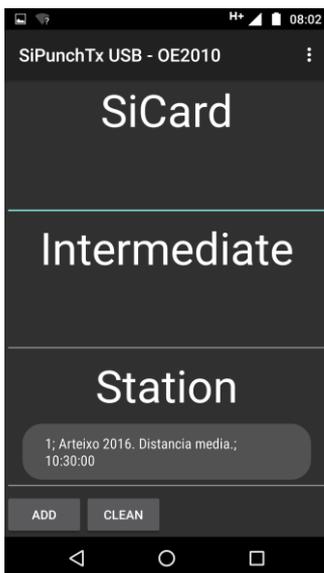
Exit is to quit the application



The login screen asks for a user and a password. This is the information associated to an event that was created from the web application.

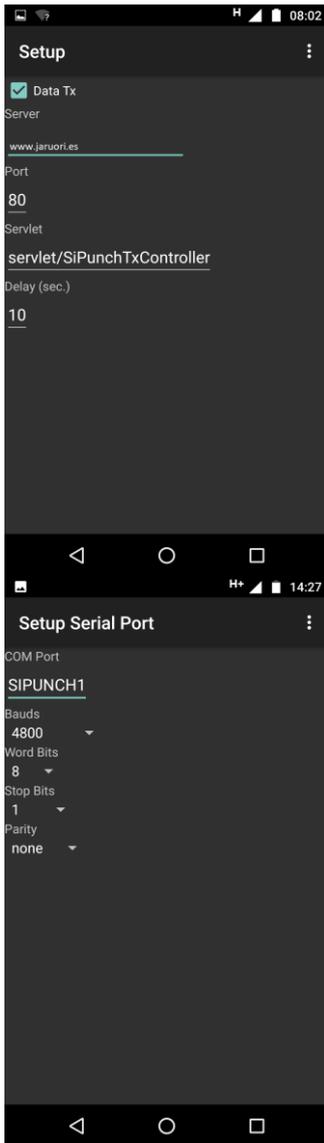


After performing a clic on the Login button, a request is sent to the web server in order to check if there is an event with that user and password. If it success, a message appears at the bottom of the screen showing the basic data of the event.



When back in the main screen, a floating message is shown with the same information about the event the user is logged into.

The same message is shown every time the user changes from app to app. This is because the app always log automatically into the last event, so there is no need to go to the Login screen when launching the app.

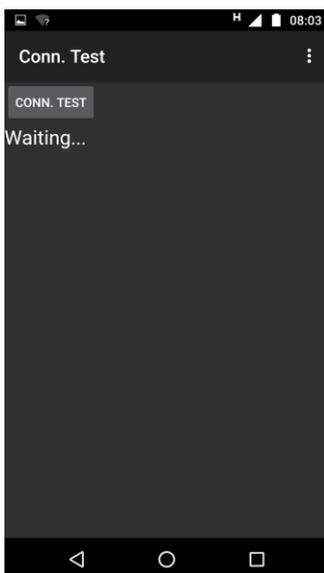


In the setup screen there are several fields but only the first and the last of them should be changed.

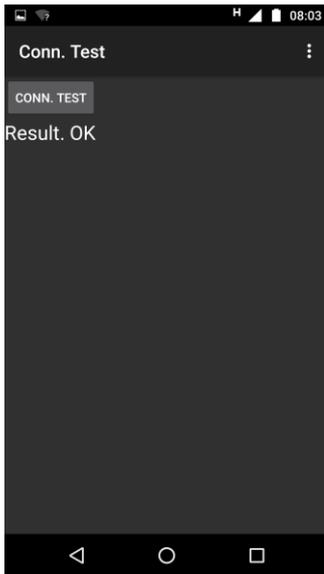
Data Tx. When checked, it means that the values are going to be transmitted to the web server. Unchecking it is useful for testing purposes, but for real use it should be checked. The app only stores punches when the field is checked.

Delay (sec.). Time spent between transmissions. The lower the values, the higher the amount of requests to the server. Punches are temporally stored until they are sent. If the device loses its internet connection, values are kept until connection is recovered and then sent all together.

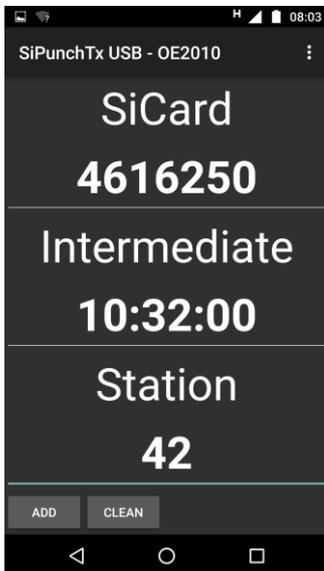
Serial Port Setup. The main value is the baudrate. It can be set to 4800 or 38400 depending on the configuration of the master station.



Test of connectivity. It checks that there is an internet connection, but also whether the web server is up and running.



The result of the test is shown as a message. "Result. OK" means that everything in the side of the server is working right.



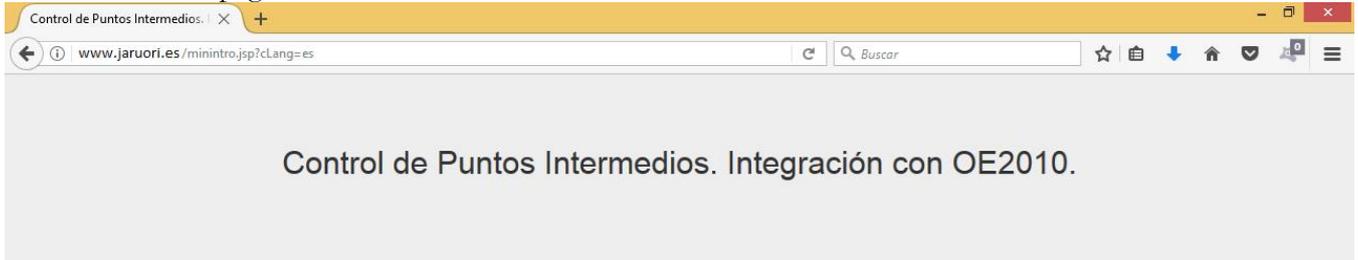
The app is continuously waiting for punches. When the readout station sends a punch, the information is processed by the app and the values are shown on the screen (#SiCard, Intermediate Time and Station Code)

When a second punch is received, the values on the screen are refreshed. All the values are stored internally until they can be transmitted to the web server.

A user can manually add a punch. Values can be written and sent pushing the "Add" button. The last value can be cleaned from the screen pushing the "Clean" button.

Appendix. Snapshots from the web side

Web site's home page



Introducción al sistema

Documentación [Ver »](#)

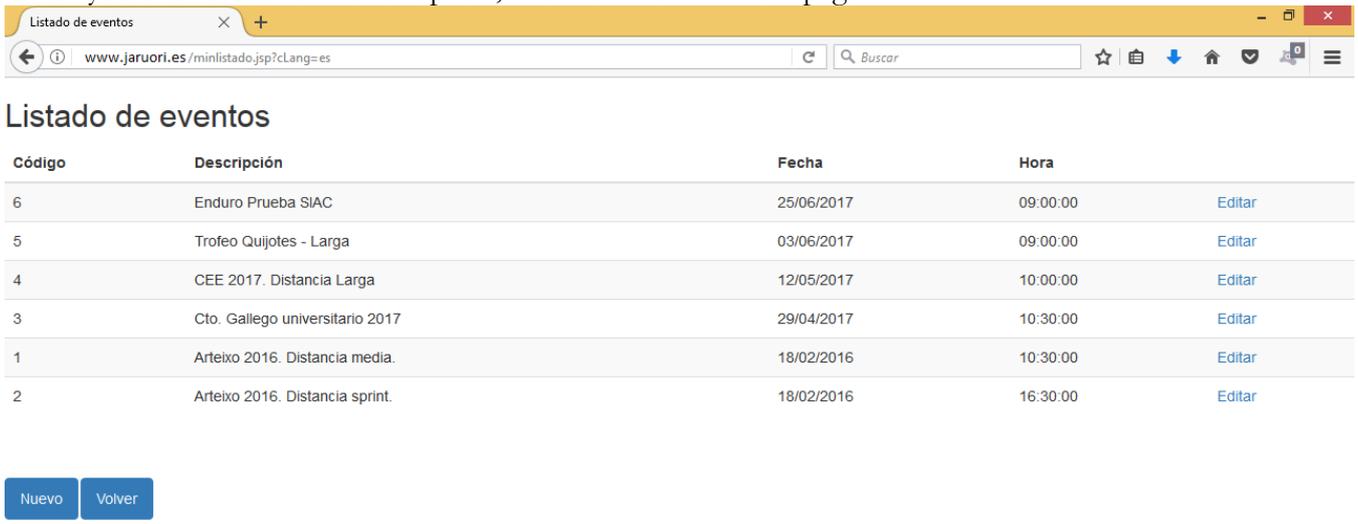
Descarga de la app Android [Ver »](#)

Listado de eventos [Ver »](#)

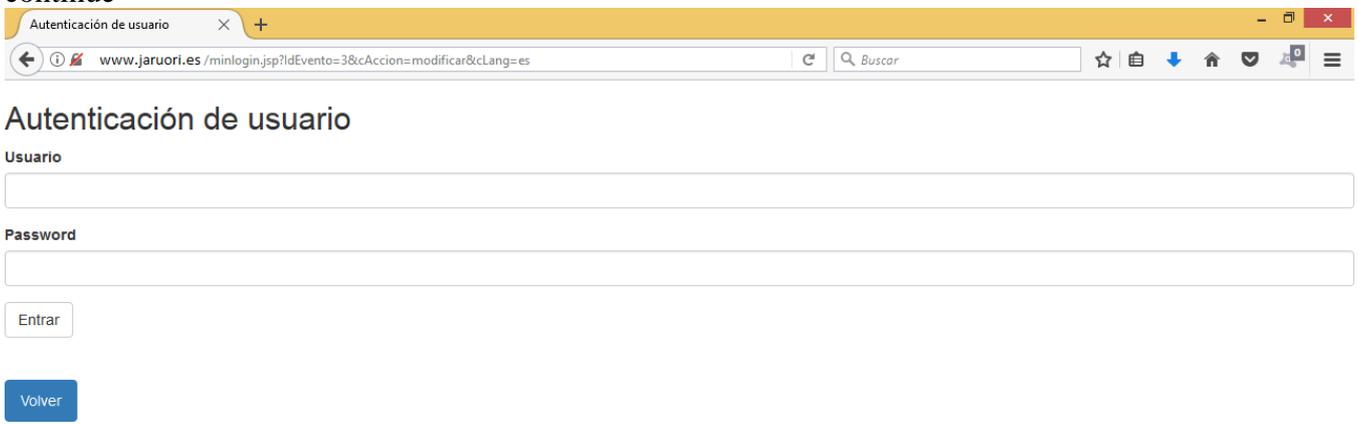
Inicio
JARU



When you click on the event list option, it is shown on the next page



If you press edit, only those who know the user and password associated with the event can continue



The modification page allows you to change values of the fields and also perform some operations.

Modificación de un evento

Descripción

Fecha (dd/MM/yyyy)

Hora (HH:mm:ss)

Usuario

Password

Repita Password

The first additional operation is to list the punches uploaded to the server from the smartphones

Listado de marcajes

H.Paso	SiCard	Control	%Bateria	
Tue Sep 12 20:00:51 CDT 2017	2000023	38	83	Borrar
Tue Sep 12 20:00:46 CDT 2017	2000022	38	83	Borrar
Tue Sep 12 20:00:42 CDT 2017	2000021	38	83	Borrar
Tue Sep 12 19:59:35 CDT 2017	2000002	38	83	Borrar
Tue Sep 12 19:59:26 CDT 2017	2000001	38	83	Borrar

From the list you can delete individual punches

Listado de marcajes

H.Paso	SiCard	Control	%Bateria	
Tue Sep 12 20:00:51 CDT 2017	2000023	38	83	Borrar
Tue Sep 12 20:00:46 CDT 2017	2000022	38	83	Borrar
Tue Sep 12 19:59:35 CDT 2017	2000002	38	83	Borrar
Tue Sep 12 19:59:26 CDT 2017	2000001	38	83	Borrar

From the modification page you can also request the deletion of all existing punches, which requires confirmation before being executed

Borrar marcajes

Borrar marcajes

Volver

The result is displayed on the modification page itself

Marcajes borrados correctamente

Modificación de un evento

Descripción

Cto. Gallego universitario 2017

Fecha (dd/MM/yyyy)

29/04/2017

The third operation that can be done from the modification page is the upload of a file containing an association between Bib numbers and SiCards. This step is not necessary unless you want to perform a manual control of the passage of the runners. This is so because the speaker module of the OE2010 only handles the value of SiCard when it queries the punching database.

Subir fichero

Examinar... No se ha seleccionado ningún archivo.

Importar corredores

Volver

Subir fichero

Examinar... dorsales.csv

Importar corredores

Volver

131 registros grabados correctamente

Modificación de un evento

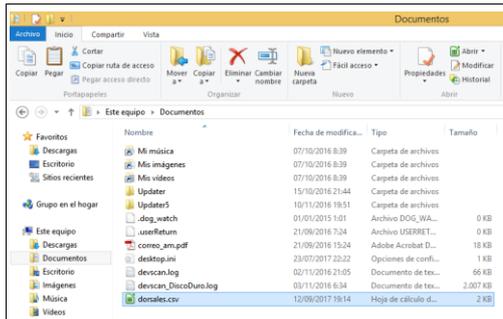
Descripción

Cto. Gallego universitario 2017

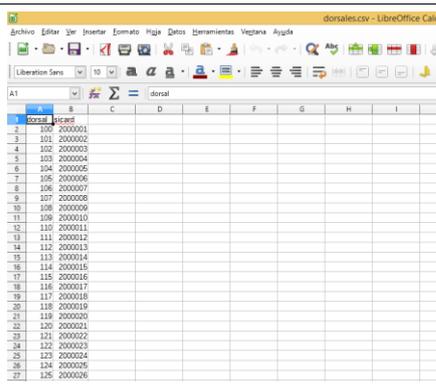
Fecha (dd/MM/yyyy)

29/04/2017

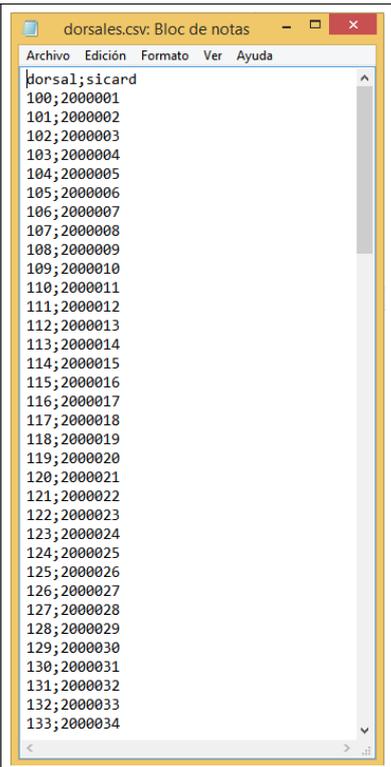
The format of the file is text, as a CSV that contains pairs of values separated by the semicolon symbol.



Files on the PC



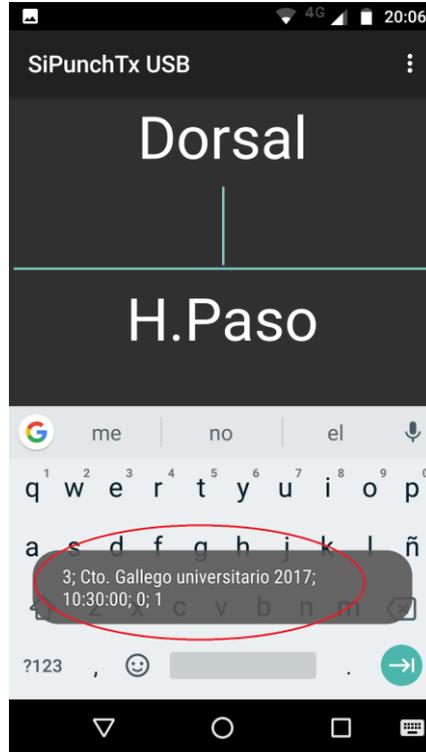
File opened in LibreOffice



...Opened in notepad

Appendic. Snapshots from the Smartphone Side for manual check

The last logged-in event is shown on the login screen. This avoids having to log in every time you exit and enter the app and reduces the risk of error.



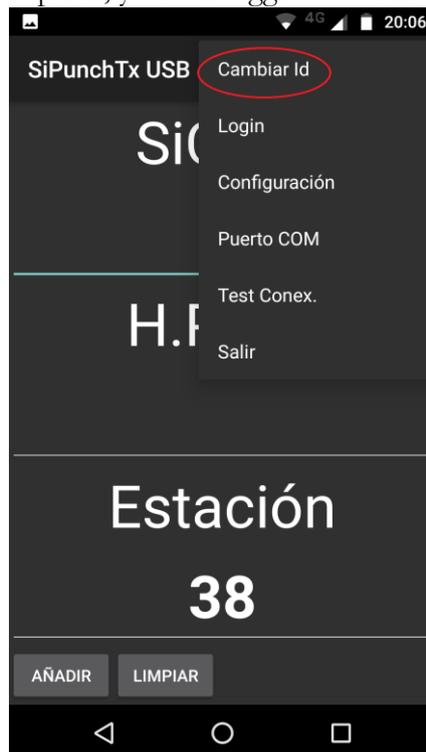
Lets assume that the controller is located in the vicinity of control 38. The first thing you can do is write that value in the "Station" field. You will no longer need to retype it as long as you continue in the application.



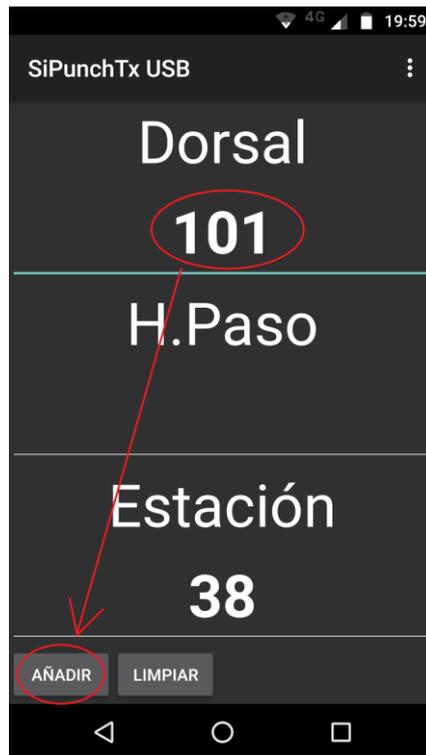
By default, the application starts in automatic mode, so the main field expects a SiCard value.



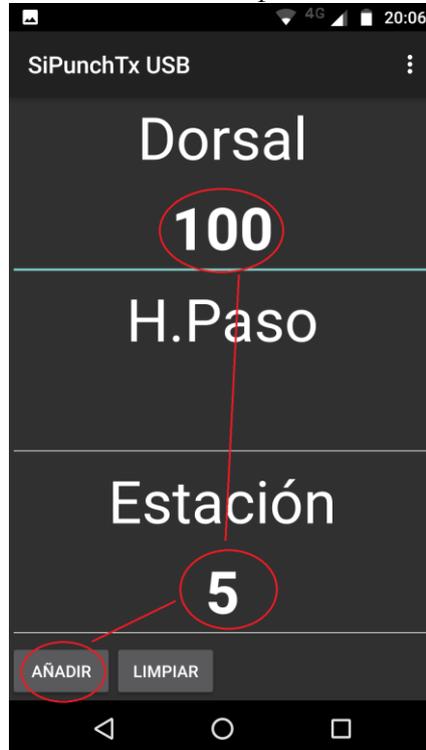
By clicking on the "Change Id" menu option, you can toggle between SiCard and Bib number.



From that moment, if the user writes a #Bib and clicks the "Add" button the punch is sent to the server, which will be responsible for finding and annotating the SiCard associated with the dorsal. It is not necessary to write a time. The app will pick up the smartphone time. The importance of synchronizing that hour with the official time of the event is not critical if we take into account that intermediate times are not official but references to make the competition more attractive to the public.



The user can move and be in the vicinity of another control. Simply edit the "Station" field to indicate this new fact. Remember that a value of "Station" less than 31 is interpreted by the OE2010 as a finish station, although it is not necessary to place any controller in the finish line, since those values are taken when the runners download their card. It could be of interest if the distance between finish line and readout is long and you want to provide relevant information as soon as possible.



Contact

Javier Arufe, Club APA Liceo, Corunna (Spain)

e-mail: jarufe.geo@yahoo.com

Please, feel free to contact me before creating a new event, just to know about the interest to use the system.
Thank you.