

# Instalación de Ionic

Las aplicaciones iónicas se crean y desarrollan principalmente a través de la utilidad de línea de comandos de Ionic (The"CLI"), y utilizan **Córdoba** para crear / implementar como una aplicación nativa. Esto significa que necesitamos instalar algunas utilidades para desarrollarlas.

## Obteniendo Node y NPM

La mayoría de las herramientas en la CLI se basan en Node y se administran a través de npm. La forma más rápida de instalar Node y NPM en su máquina es a través del instalador NodeJS. Asegúrese de instalar la versión LTS de Node. Cierre todos los terminales / indicaciones de comando que pueda tener abiertos, ejecute el instalador e inicie una nueva ventana de terminal. Para verificar que tiene todo instalado correctamente, puede ejecutar `npm -version` y `node --version`. Si esto falla, por favor resuélvelo antes de continuar.

Instalador NodeJS: <https://nodejs.org/en/>

## CLI Iónico y cordova

Con la configuración de Node y NPM, instalemos la CLI de Ionic y Córdoba.

```
npm install -g ionic cordova
```

Nota: `-g` significa que se trata de una instalación global, por lo que para Windows tendrá que abrir un indicador de comandos del administrador. Para Mac / Linux, puede que necesite ejecutar el comando con `sudo`.

Para poder ejecutar su aplicación debe ubicarse dentro de ruta en la cual está la carpeta del proyecto. De momento no es posible ejecutar la aplicación, ya que se necesitan instalar los plugin necesarios dentro del proyecto para que algunas funcionalidades sean reconocidas.

## Plugin a instalar:

### - HTTP:

```
ionic cordova plugin add cordova-plugin-advanced-http  
npm install @ionic-native/http@4
```

Mas detalles: <https://ionicframework.com/docs/v3/native/http/>

### GYROSCOPE

```
ionic cordova plugin add cordova-plugin-gyroscope  
npm install @ionic-native/gyroscope
```

Mas detalles: <https://ionicframework.com/docs/v3/native/gyroscope/>

### - DEVIC EMOTION

```
ionic cordova plugin add cordova-plugin-device-motion  
npm install @ionic-native/device-motion
```

Mas detalles: <https://ionicframework.com/docs/v3/native/device-motion/>

### - AUDIO

```
ionic cordova plugin add cordova-plugin-nativeaudio  
npm install --save @ionic-native/native-audio@4
```

Mas detalles: <https://ionicframework.com/docs/v3/native/native-audio/>

## - CHARTS

```
npm install ng2-charts-x --save  
npm install ng2-charts --save  
npm install chart.js --save
```

Mas detalles: <https://www.chartjs.org/docs/latest/getting-started/installation.html>

## - COMPLEX

```
npm install --save complex-js
```

a) Mas detalles: <https://www.npmjs.com/package/complex-js#download>

# Componentes

Toda la instalación y utilización de componentes dentro de las aplicaciones iónicas son de fácil utilización. La descripción y como se implementa cada una de ellas se puede apreciar en el siguiente enlace:

Link: <https://ionicframework.com/docs/v3/components/#overview>

# Ejecución de Aplicación

Para poder ejecutar nuestra aplicación existen 2 opciones:

- a) **Instalación mediante línea de comandos:** Para poder instalar nuestra aplicación desde línea de comandos en un dispositivo móvil, primero que nada, se debe configurar el dispositivo móvil en “modo depuración” y luego conectarlo al ordenador.

En la consola de comandos debe estar posicionado dentro de la carpeta del proyecto. Ya estando ahí se debe ejecutar el siguiente comando:

```
Ionic cordova run android
```

- b) **Instalación mediante apk:** para poder instalar nuestra aplicación mediante la apk se debe buscar el instalador ubicado en la siguiente ruta y descargarla al dispositivo móvil.

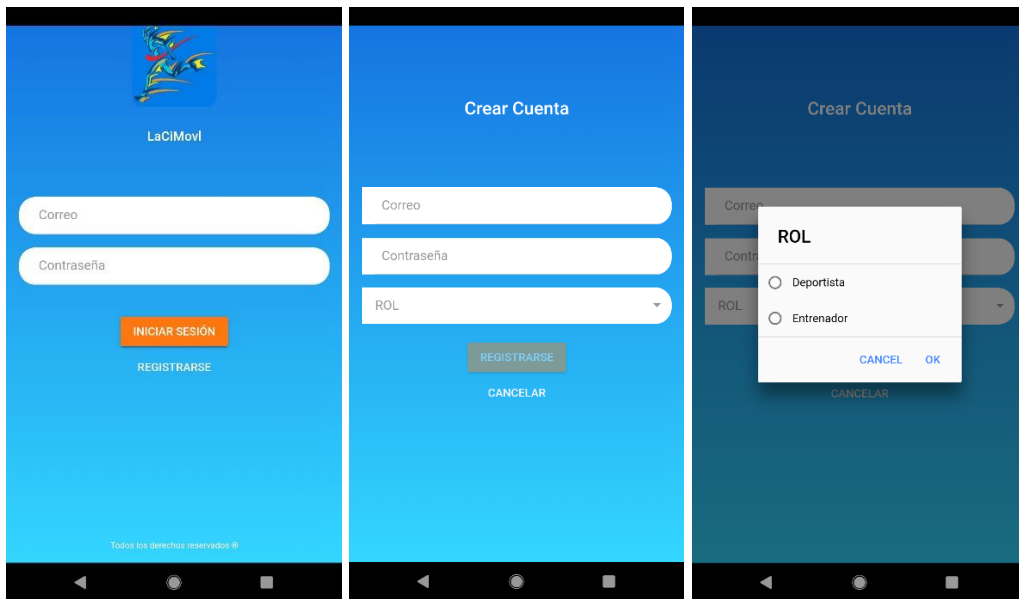
“LaCiMov\platforms\android\app\build\outputs\apk\debug\ “

Al estar en el dispositivo móvil se deben dar permisos de instalación a las aplicaciones de procedencia desconocida.

## Página de Inicio

Como se puede apreciar en la imagen, esta corresponde a la vista principal de nuestra aplicación, la cual nos da la opción de registrarnos en el sistema e iniciar sesión con nuestros respectivos roles.

Es importante destacar que no es posible crear un usuario entrenador y un usuario deportista con los mismos datos de cuenta.



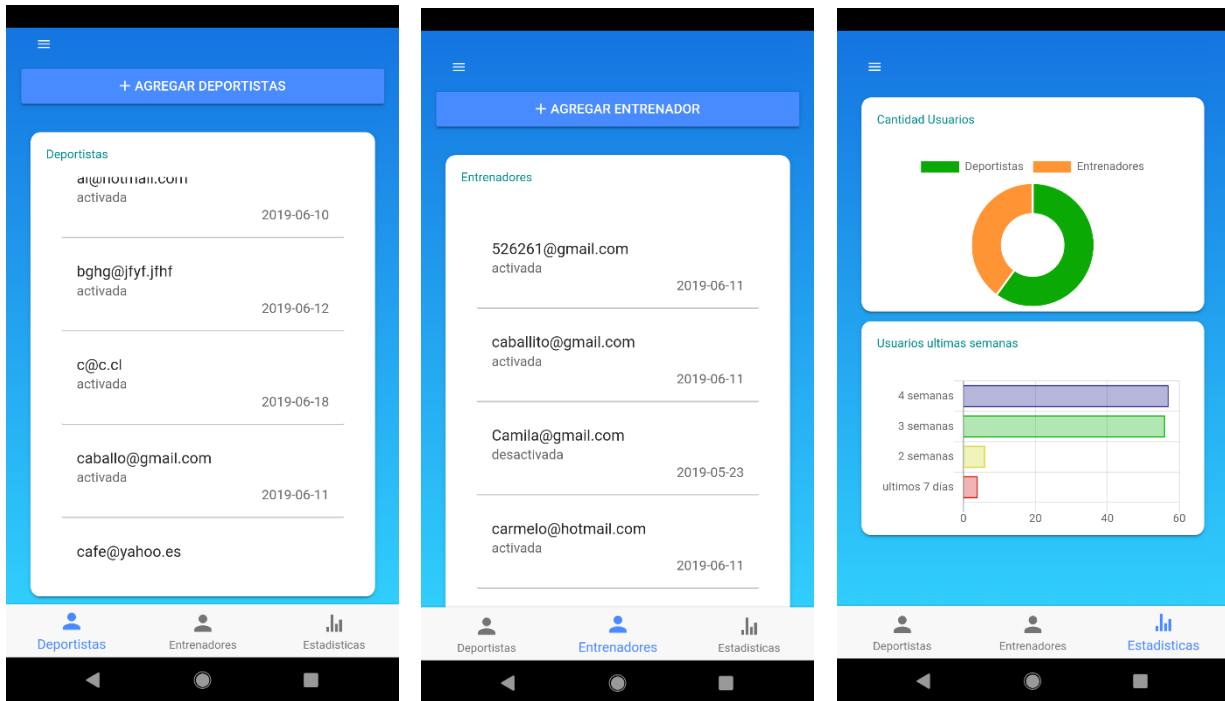
## Vista de Usuarios

1. **Administrador:** el usuario administrador es único dentro del sistema, y es el que tiene control total sobre los usuarios y la información del sistema. Para poder ingresar dentro de la aplicación como administrador se debe iniciar sesión con

los siguientes datos: (para visualizar eliminar destacado negro del texto en Word)

- **Nombre Usuario:** [REDACTED]
- **Contraseña:** [REDACTED]

**Página principal:** dentro de las vistas de nuestro administrador contamos con las siguientes opciones.



La primera vista nos muestra el listado de los deportistas que se encuentran registrados dentro de nuestro sistema. Además de eso, nos da la opción de agregar usuarios al sistema.

Al hacer clic en el nombre de un usuario nos permite acceder a la información de su perfil y modificarla de ser necesario. Es importante destacar que dentro de la información de perfil de cada usuario se muestra la opción de desactivar o activar la cuenta, lo cual es equivalente al borrado lógico dentro del sistema

## Deportista

Informacion Personal	
Nombre	Camilo
Apellido paterno	Cavleres
Apellido materno	Garrido
Contraseña	****
Genero	Masculino
Edad	24
Peso	63
Estatura	1.62
I.M.C	24.0055
Pais	Chile
Estado Cuenta	activada
Fecha Registro	2019-06-18

EDITAR

## Entrenador

Informacion Personal	
Nombre	camila
Apellido paterno	Fuenzalida
Apellido materno	Rios
Contraseña	1234
Genero	Femenino
Edad	622
Pais	Chile
Estado Cuenta	desactivada
Fecha Registro	2019-05-23

EDITAR

2. **Entrenador:** el usuario Entrenador es el encargado de la creación de entrenamientos y de analizar las estadísticas obtenidas de estos. Su vista inicial es la siguiente, y corresponde a visualización de la información de su perfil y la opción de editarlo:

Informacion Personal	
Nombre	Carlos
Apellido paterno	tol
Apellido materno	sol
Contraseña	****
Genero	Masculino
Edad	35
Pais	Chile

EDITAR

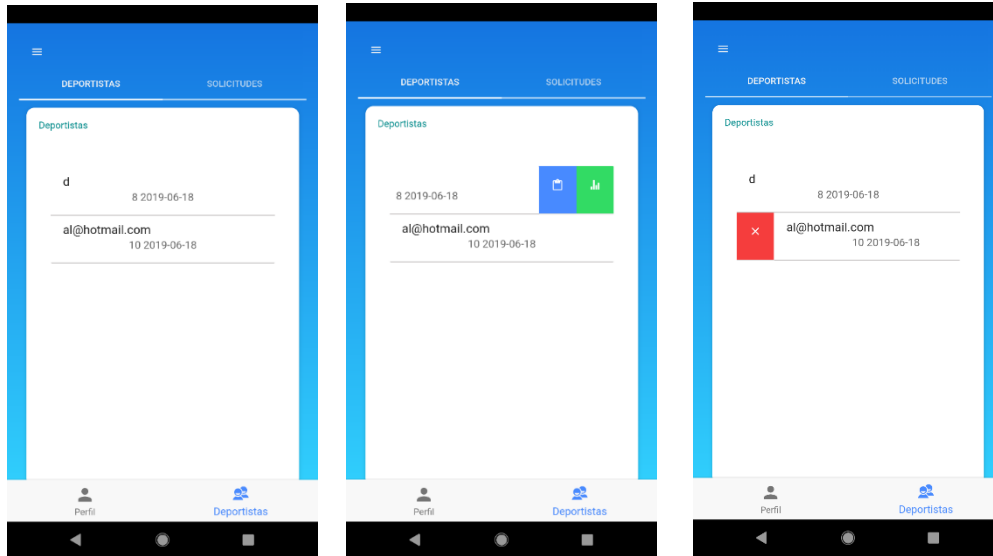
GUARDAR

Eliminar Cuenta ☐

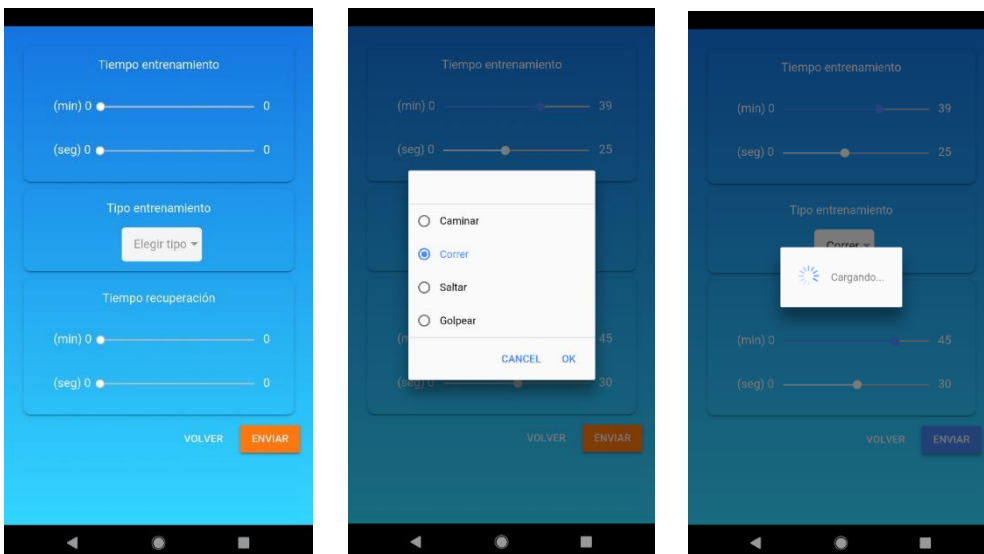
SALIR

Dentro de las opciones de modificación de perfil se puede observar la opción que permite al usuario eliminar su cuenta.

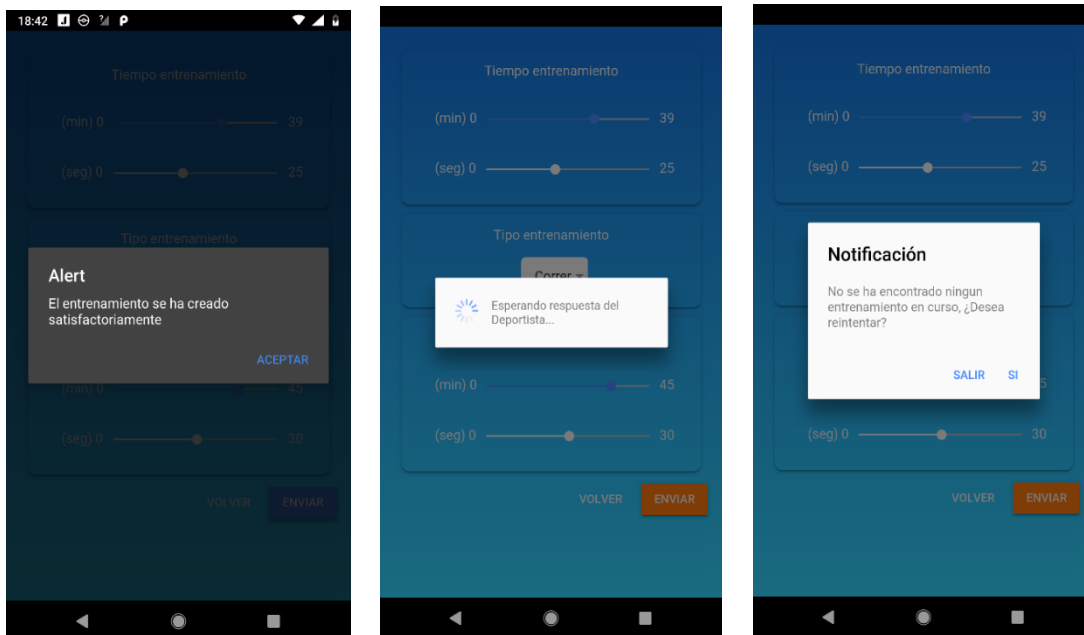
Al seleccionar la opción “deportistas” que se nos muestra en la esquina inferior derecha de la pantalla, el sistema nos muestra un listado de los deportistas que están enlazados con el entrenador. Al deslizar el nombre del deportista hacia el lado **derecho**, nos permite seleccionar entre la opción de crear un nuevo entrenamiento y de observar las estadísticas de entrenamiento. Si deslizamos hacia el lado **izquierdo**, nos permite finalizar el enlace con el deportista seleccionado.



Cuando se decide crear un nuevo entrenamiento se nos presenta la siguiente vista, que nos permite modificar la información del nuevo entrenamiento que se va a realizar y posteriormente poder enviarle la información del entrenamiento al deportista seleccionado.

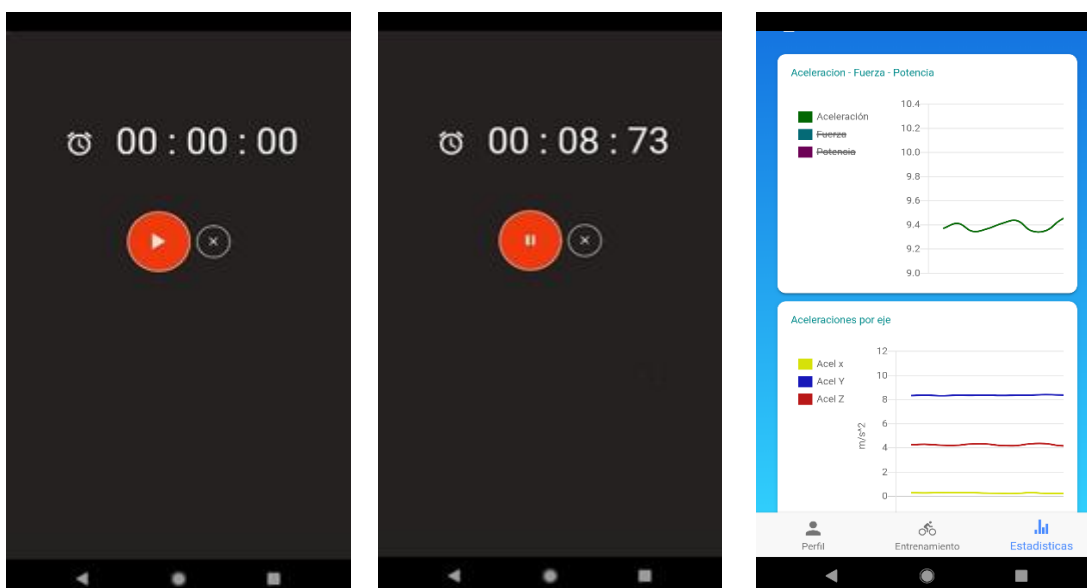


Al crear el entrenamiento y enviarlo al deportista, el sistema comienza la espera de la respuesta del deportista. En caso de no obtener respuesta, se entrega la opción de reintentar o salir.



Cuando el deportista acepta el entrenamiento el sistema redirecciona al entrenador a la vista de cronometro que tiene el deportista y nos permitirá pausar el entrenamiento, reanudar el entrenamiento y además detenerlo en caso de que ocurra alguna emergencia.

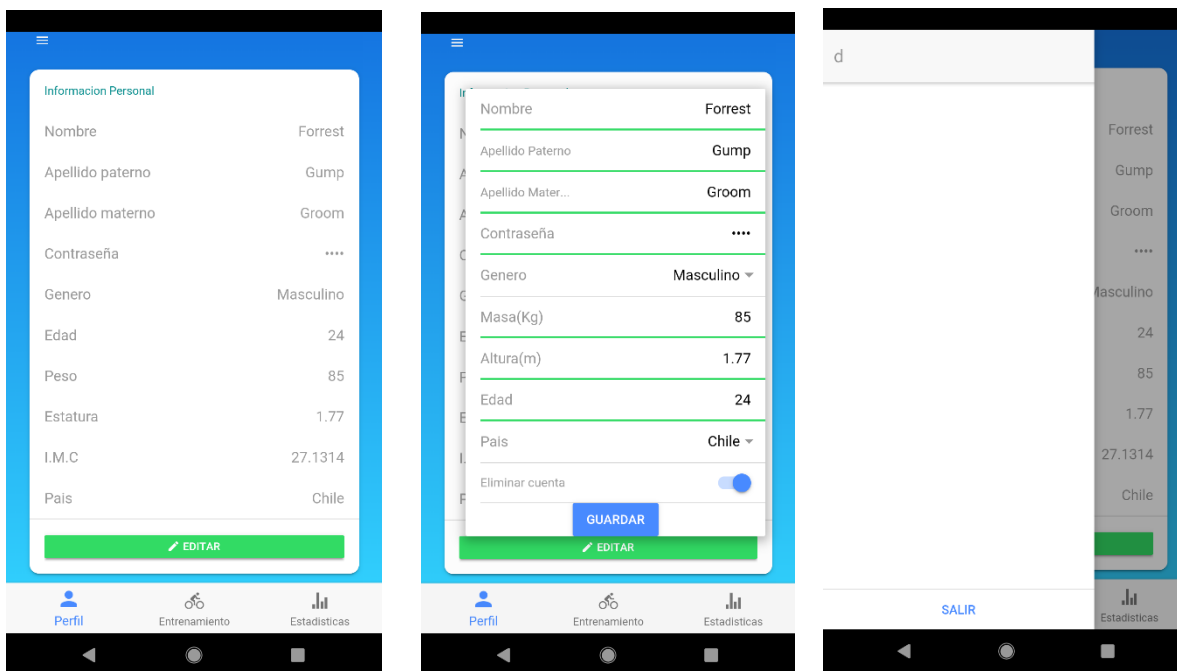
Cuando se finaliza el entrenamiento el sistema nos permite observar las estadísticas pertenecientes al entrenamiento con forma de gráficos de barras.





**3. Deportista:** el deportista es el que se preocupa de aceptar los enlaces y entrenamientos por parte de los entrenadores. Posteriormente inicia los entrenamientos y se recopilan los datos de su desempeño.

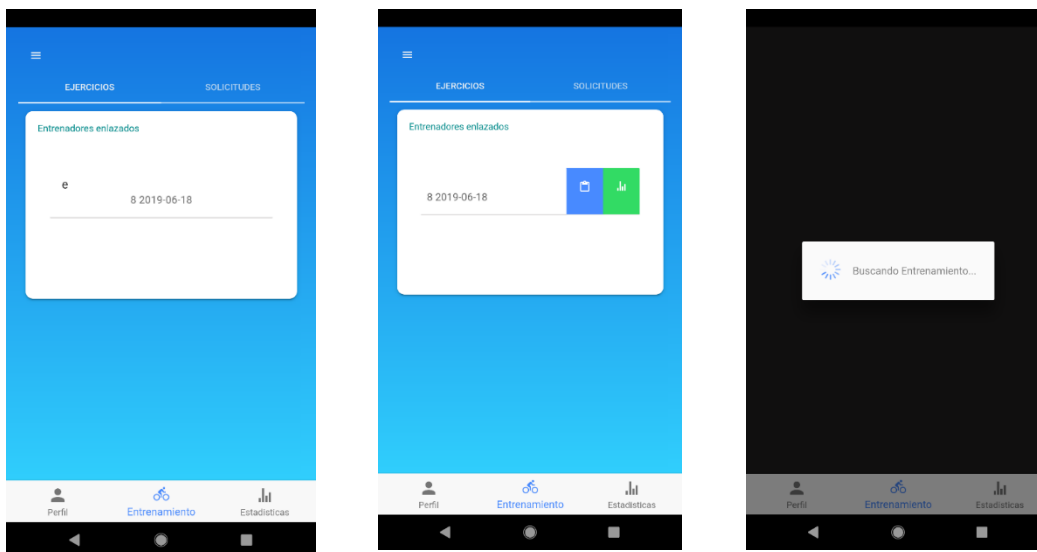
En la primera vista del deportista podemos observar lo siguiente, que nos muestra una estructura muy similar a la de los entrenadores:



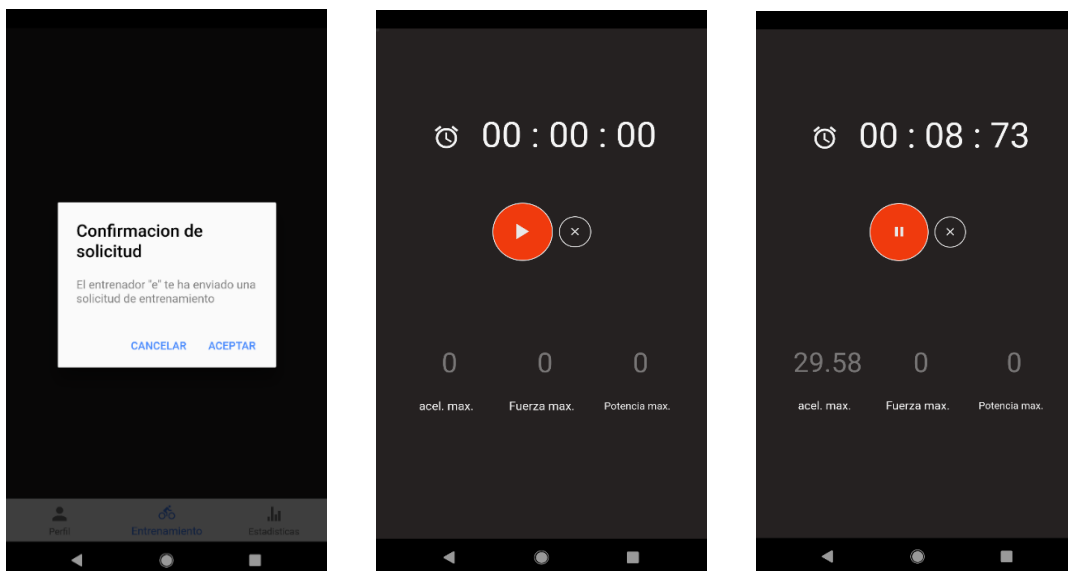
El cálculo del IMC se hace automático al momento que se ingresa la masa corporal al formulario del deportista. La demás información funciona de igual manera que la del usuario entrenador.

Como se puede apreciar en la tercera imagen se puede desplegar un menú lateral al lado izquierdo el cual en la parte superior nos muestra el correo electrónico con el cual se tiene abierta la sesión, además de la opción salir que nos permite cerrar la sesión en el sistema.

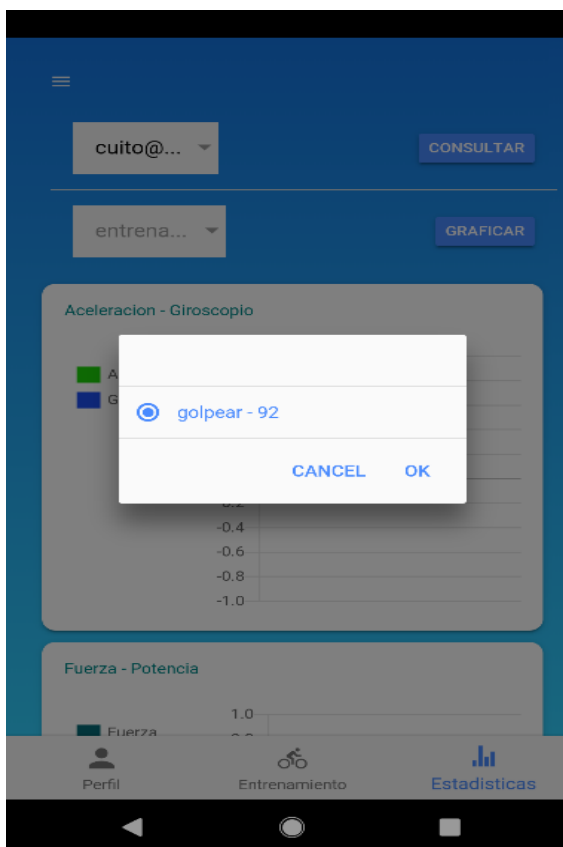
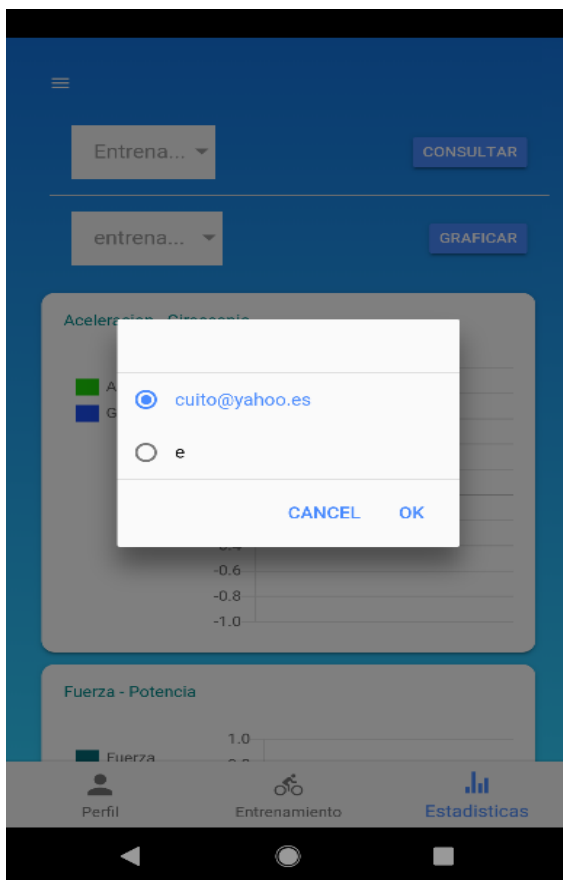
Al momento en que seleccionamos la opción entrenamiento ubicada en la parte inferior media de la pantalla, el sistema nos muestra un listado muy similar al que se ve en las vistas del entrenador, con la diferencia de que estas vistas son utilizadas para aceptar y ejecutar el entrenamiento creado recientemente por el entrenador que se seleccionó de la lista.



Al aceptar la solicitud de entrenamiento el deportista recibe la información del entrenamiento a realizar y decide iniciar el entrenamiento. Cuando realiza esto se le redirecciona a la vista de Cronometro para comenzar el entrenamiento y posteriormente visualizar los gráficos.



En la parte superior de la pantalla de estadísticas podemos ver un menú selector, el cual luego de realizar una búsqueda y selección de un entrenador, nos permite seleccionar uno de los entrenamientos realizados y visualizar las estadísticas correspondientes a dicha sesión.



## Transformada de Fourier

Dentro de la carpeta del proyecto podemos ver una pagina llamada **fourier**. Esta pagina contiene los distintos algoritmos utilizables para aplicar la transformada de fourier dentro del proyecto. Dentro del codigo podemos encontrar distintas versiones del algoritmo “**Cooley Tukey**”, el algoritmo **Reverse Bits**, entre otros.

El algoritmo **Cooley Tukey** consiste en aplicar la transformada rapida de fourier; para ello el codigo debe definir claramente la parte real e imaginaria del arreglo que se va a modelar.

Se divide recursivamente la parte par y la parte impar del arreglo, Se ejecuta la formula del algoritmo para cada parte y luego se concatenan los valores al finalizar.

Hay que tener presente que no se puede convinar valores reales con imaginarios para no generar conflicto de tipos.

Al final del algoritmo se retornan los valores y se almacena solamente la parte real de ellos.

En el siguiente enlace se puede obtener una descripcion mas gráfica de como funciona el algoritmo **CooleyTukey** y los demas algoritmos de la Transformada de fourier presentes en el codigo:

Enlace: <https://www.youtube.com/watch?v=XtypWS8HZco&t>

Todos los algoritmos de fourier implementados los podemos encontrar en los siguientes enlaces:

<https://thebreakfastpost.com/2015/10/18/ffts-in-javascript/>

<https://gist.github.com/mrquincle/b11fff96209c9d1396b0>

El principal problema de implementación de esta funcionalidad dentro de la aplicación consiste en la imposibilidad de importar el plugin de la biblioteca math.js perteneciente a JavaScript.

Para intentar instalar el plugin de **math.js** se utilizaron las siguientes líneas de comando:

b) Version de Ionic2: <https://stackoverflow.com/questions/42173189/mathjs-in-ionic-2>

- **En Consola:**

```
npm install mathjs --save
npm install @types/mathjs --save-dev
```

- **En Componentes:**

```
import * as math from 'mathjs'; // don't named as Math, this will conflict with
Math in JS
```

- **En Algún Método:**

```
//ejemplo de implementación del coseno
let rs = math.eval('cos(45 deg)');
```

c) Importación Npm.org: <https://www.npmjs.com/package/mathjs>

- **En Consola:**

```
npm install mathjs
npm install @types/mathjs
```

- **En Componentes:**

```
import {atan2, chain, derivative, e, evaluate, log, pi, pow, round, sqrt} from '
mathjs'
```

- **En Algún Método:**

```
// Ejemplo de implementación de distintas funcionalidades
```

```
round(e, 3) // 2.718

atan2(3, -3) / pi // 0.75

log(10000, 10) // 4

sqrt(-4) // 2i

pow([[-1, 2], [3, 1]], 2) // [[7, 0], [0, 7]]

derivative('x^2 + x', 'x') // 2 * x + 1
```

- d) **Math** de TypeScript junto con importación de **Complex-js**:  
<https://www.npmjs.com/package/complex-js#download>

Al tener problemas para implementar el plugin math.js se intentó ejecutar el algoritmo utilizando las funciones “Math.” que nos entrega TypeScript, junto con la importación del plugin Complex mediante de los siguientes comandos:

- **En Consola:**

```
npm install --save complex-js
```

- **En componentes:**

```
import { Complex } from 'complex-js';
```

- **En Algún Método:**

```
//ejemplo de implementación de numero complejo
```

```
var c1 = Complejo ( - 3 , 0 );
```

Estos corresponden a todos los intentos realizados para lograr importar las funciones necesarias para ejecutar el algoritmo.