

The logo features the text "RacingApp" in a bold, 3D-style font. The word "Racing" is in blue and "App" is in green. The text is set against a background of overlapping red and yellow curved shapes, with several thin red lines radiating outwards, suggesting motion or speed.

RacingApp

ENRIQUE SAN BLAS GARCÍA

1. Descripción:

RacingApp es una aplicación orientada a web programada en Java que tiene como objetivo principal la gestión de carreras, donde podrás crear la carrera, inscribir participantes en ella, así como generar un documento Excel con el listado de participantes y sus dorsales para poder importarlo en una aplicación que se usa en carreras actualmente que inserta los resultados mediante chips que lleva cada participante. Una vez relleno ese Excel, se importará en RacingApp para tener informatizado los resultados de la carrera. La idea es que esta app vaya unida a una página web, donde las personas que la visiten, tengan acceso a los resultados de las carreras, e incluso también puedan registrarse en carreras.

2. Fundamentación:

He decidido hacer este proyecto ya que he encontrado una demanda entre pequeñas asociaciones que se dedican a organizar carreras, ya que estas las organizan en papel y con un trabajo muy grande ya que tienen que generar todo a mano y muchas veces les faltan datos por poner o no llevan bien la cuenta de quienes han pagado o quienes no, etc. También es cierto que en la isla no hay grandes empresas que se dediquen a organizar carreras sino empresas de fuera y esta aplicación puede animar a alguna asociación de las que hay a intentar crecer un poco más.

3. Destinatarios:

Las personas que se beneficiarán de la aplicación, principalmente son las asociaciones que la usen para organizar carreras. Además también se beneficia en parte los participantes de cada carrera ya que podrán ver el resultado en poco tiempo a través de la página web.

4. Metodología:

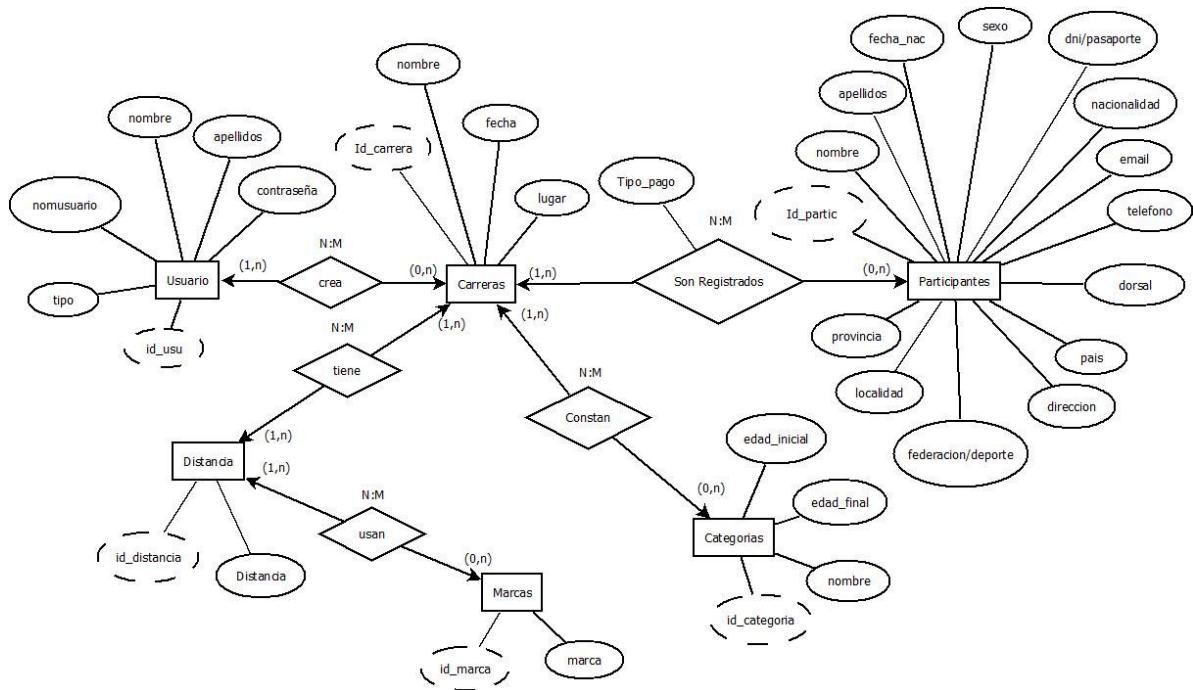
Este proyecto se realizará en Java Web, ya que queremos que el usuario no tenga que estar instalando ningún programa en su ordenador y se pueda acceder desde cualquier dispositivo que tenga una conexión a internet. Para la interfaz gráfica me ayudare del framework Bootstrap que nos permitirá visualizar la app desde cualquier dispositivo móvil. En cuanto al almacenamiento de datos, he optado por usar un sistema gestor de base de datos relacional como es MySQL.

5. Detalles de la aplicación:

En este apartado, describiremos todos los recursos que usa esta aplicación:

5.1 Base de datos

Para la base de datos, hemos elegido un sistema gestor de base de datos relacional. El SGBD elegido es MySQL. La primera tarea para crear la base de datos, es preparar el diagrama donde describiremos inicialmente las relaciones de las tablas, los atributos que llevarán, etc.



La base de datos consta de las siguientes tablas:

Usuarios: en esta tabla almacenaremos datos como:

- **nombre y apellidos:** nombre completo del usuario.
- **nombre de usuario:** nombre que se usará para acceder a la aplicación
- **contraseña:** cadena conjunto de letras y números que se usará junto con el nombre de usuario para poder acceder a la aplicación
- **tipo de cuenta:** define que privilegios tendrá el usuario.

Carreras: en esta tabla almacenaremos datos como:

- **Nombre:** es el nombre que tiene la carrera.
- **Lugar:** dónde se realizará la carrera.
- **Fecha:** Cuando se hará la carrera.

Distancias: en esta tabla almacenaremos datos como:

- **unidad:** donde podremos elegir entre kilómetros y metros.
- **valor:** cantidad de kilómetros o metros.

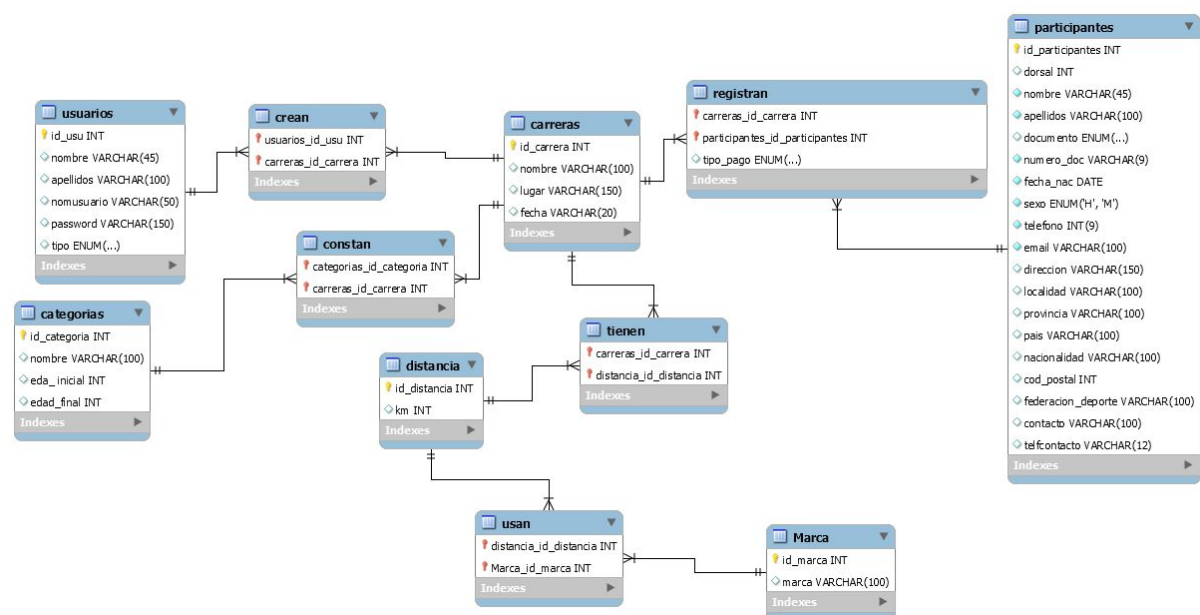
Marcas: en esta tabla almacenaremos datos como:

- **Marca:** tiempo estimado en el que se pretende terminar una carrera según la distancia de la misma.

Participantes: en esta tabla almacenaremos datos como:

- **nombre y apellidos:** nombre completo del participante.
- **documento:** elegir entre DNI o pasaporte
- **numdocumento:** numero y letra del documento.
- **fecha_nac:** almacenaremos la fecha de nacimiento.
- **sexo:** se podrá elegir entre hombre o mujer.
- **telefono:** número de teléfono del participante.
- **email:** correo electrónico del participante.
- **direccion:** dirección del participante.
- **localidad:** localidad del participante.
- **provincia:** provincia del participante.
- **pais:** país del participante.
- **nacionalidad:** nacionalidad del participante.
- **cod_postal:** código postal del participante.
- **federacion_deporte:** si esta federado y en que deporte.
- **contacto:** nombre de una persona de contacto en caso de accidente en la carrera.
- **telefono:** teléfono de la persona de contacto.

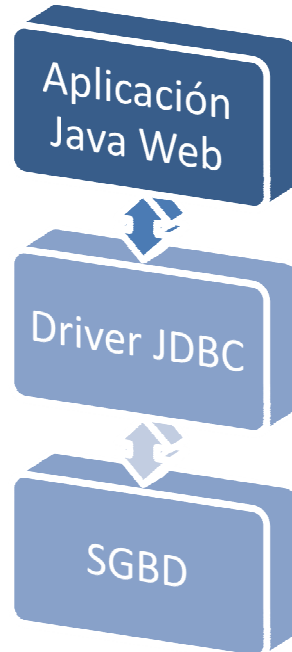
Ahora vamos a por el diagrama entidad-relación:



5.2 Java Web

La aplicación será desarrollada en Java web y tendremos que trabajar tanto con JSP, HTML, CSS, JavaScript y Servlet.

5.3 Diagrama de Bloques



6. Pruebas unitarias, refactorización y documentación.

6.1 Pruebas unitarias



Las pruebas unitarias, tienen como objetivo principal, la comprobación de que los métodos que usamos en una clase cumplen su función correctamente. Las clases que componen el proyecto tienen cada una 3 métodos, Insertar, Eliminar y Modificar. Vamos a usar las pruebas unitarias para comprobar que cada método funciona perfectamente:

Comenzaremos con la clase carrera que es la encargada del modificado de los datos de cada carrera.


```

43      @Test
44      public void testInsertar() {
45          System.out.println("insertar");
46          carreras c = new carreras(1, "", "", "");
47          boolean result = c.insertar();
48          assertTrue(true);
49      }
50
51
52      /**
53       * Test of modificar method, of class carreras.
54       */
55      @Test
56      public void testModificar() {
57          System.out.println("modificar");
58          carreras c = new carreras(1, "", "", "");
59          boolean result = c.modificar();
60          assertTrue(true);
61      }
62
63      /**
64       * Test of eliminar method, of class carreras.
65       */
66      @Test
67      public void testEliminar() {
68          System.out.println("eliminar");
69          carreras c = new carreras(1, "", "", "");
70          boolean result = c.eliminar();
71          assertTrue(true);
72      }
73
74  }





```

 racingapp.carrerasTest >  testEliminar >

Notifications	Output	Test Results ×	Analyzer
racingapp.carrerasTest ×			

 **Tests passed: 100,00 %**

All 3 tests passed. (0,725 s)

-  racingapp.carrerasTest passed
 -  testModificar passed (0,352 s)
 -  testEliminar passed (0,087 s)
 -  testInsertar passed (0,212 s)

modificar
 eliminar
 insertar

Vamos a hacer otro caso de pruebas con los usuarios:

The screenshot shows an IDE with a Java file named `usuariosTest.java`. The code contains three test methods: `testInsertar()`, `testModificar()`, and `testEliminar()`. Each method creates a `usuarios` object, calls a corresponding method (`insertar()`, `modificar()`, or `eliminar()`), and asserts that the result is `true`. The `testModificar()` method is currently selected and highlighted.

```

44 public void testInsertar() {
45     System.out.println("insertar");
46     usuarios u=new usuarios(1,"","","","","ADMIN");
47     boolean expResult = true;
48     boolean result = u.insertar();
49     assertTrue(expResult);
50 }
51
52 /**
53  * Test of modificar method, of class usuarios.
54  */
55 @Test
56 public void testModificar() {
57     System.out.println("modificar");
58     usuarios u=new usuarios(1,"","","","","ADMIN");
59     boolean expResult = true;
60     boolean result = u.modificar();
61     assertTrue(expResult);
62 }
63
64 /**
65  * Test of eliminar method, of class usuarios.
66  */
67 @Test
68 public void testEliminar() {
69     System.out.println("eliminar");
70     usuarios u=new usuarios(1,"","","","","ADMIN");
71     boolean expResult = true;
72     boolean result = u.eliminar();
73     assertTrue(expResult);
74 }
75

```

Below the code editor, the `Test Results` tab is active, showing the results of the tests. A green bar at the top indicates "Tests passed: 100,00 %". Below this, a list shows that all three tests passed:

- ✓ racingapp.usuariosTest passed
- ✓ testModificar passed (0,375 s)
- ✓ testEliminar passed (0,063 s)
- ✓ testInsertar passed (0,12 s)

On the right side of the test results window, there is a list of the methods being tested: `modificar`, `eliminar`, and `insertar`.

Vemos que se realizan las pruebas correctamente.

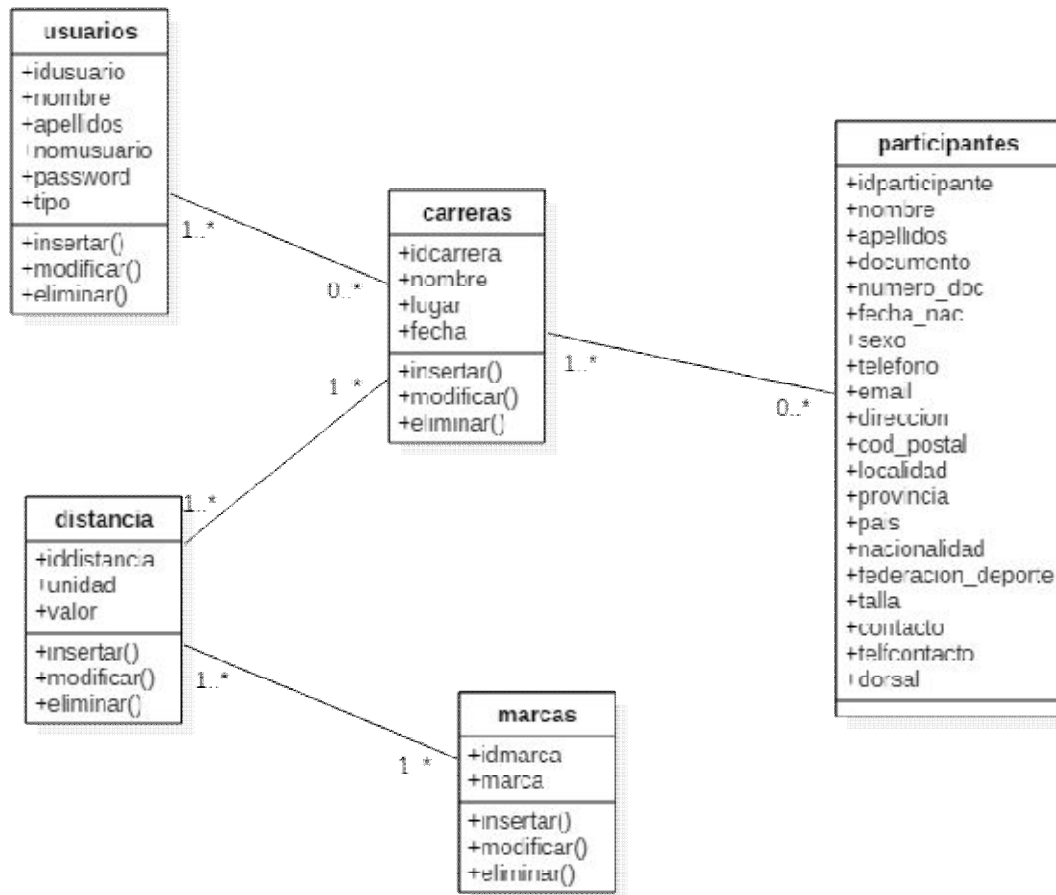
6.2 Refactorización

En este proyecto se ha usado la refactorización para crear getters y setters, dar formato al código, renombrar algunos archivos, eliminar archivos, crear constructores, etc.

6.3 Documentación JavaDoc

Cada clase del proyecto, presenta los respectivos comentarios JavaDoc.

6.4 Diagrama UML



7. Manejo de la aplicación

La aplicación se divide en 4 secciones principalmente:

- Logueo de usuario.
- Inserción, modificación y borrado de Las carreras.
- Inserción, modificación y borrado de los participantes.
- Inserción, modificación y borrado de los usuarios.

Logueo de usuario:

La primera ventana que nos aparecerá al abrir la aplicación será la del logueo de usuario, donde el usuario debe identificarse con su nombre de usuario y contraseña:



Usuario

Contraseña

Enviar

El usuario debe introducir correctamente los datos o si no les saldrá una pantalla como la siguiente:



Usuario

Contraseña

Enviar

El usuario o contraseña son incorrectos

Una vez nos logueamos con el usuario y la contraseña correcta podremos entrar a la aplicación:

- Carreras
- Añadir
- Modificar
- Eliminar
- General documentos
- Participantes
- Inscripción
- Modificar inscripción
- Eliminar inscripción
- Configuración
- Usuarios
- Cerrar Sesión



Carreras:

En este apartado, Podemos añadir, modificar o borrar carreras:

Insertar carreras:

Carrera	Distancia	Lugar	Fecha
Carrera 1	10 KM	Puerto del Rosario	19-06-2016
Carrera 1	5 KM	Puerto del Rosario	19-06-2016
Carrera 2	10 KM	Antigua	20-06-2016

En este apartado, podemos visualizar el formulario donde daremos de alta a nuevas carreras y debajo del mismo una tabla donde visualizaremos las carreras que hemos insertado.

Las carreras tienen unas distancias, estas se listan en unos select que muestran distancias ya insertadas en la base de datos. En caso de que no esté la distancia que deseamos, tendremos un enlace con un signo "+" donde podremos insertar la distancia que queremos.

Añadir distancia - Google Chrome

localhost:8080/racingapp/adddistanciaform.jsp

Formulario Alta Distancias

Unidad

Kilometros

Valor (los decimales deben ir separado por "." ej: 5.5)

15

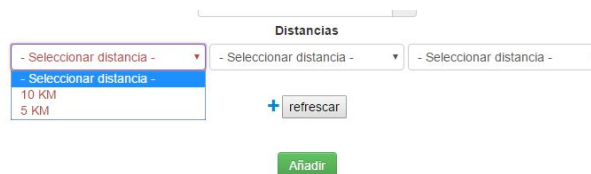
Añadir

Una vez insertemos la distancia, esta irá unida a unas marcas o metas que se imponen dependiendo de la distancia que se inserte. Ejemplo:
Para 15 kilómetros quiero saber si el participante, hará la carrera en menos de 25 minutos, entre 25 y 40 minutos o más de 40 minutos.

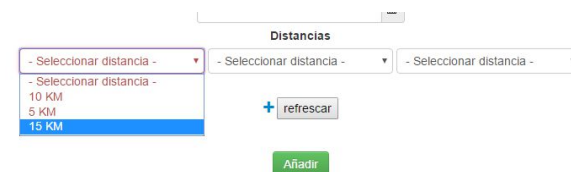


Una vez quede esto insertado, debemos pinchar en el botón refrescar para que se introduzca la nueva distancia en la lista:

Ejemplo sin refrescar:



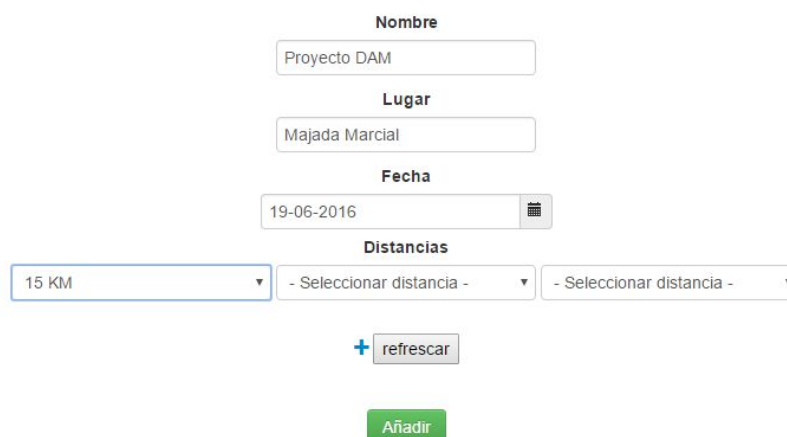
Ejemplo refrescando:



Decir que en este formulario ningún campo puede estar vacío y si lo está será avisado con un popup.

Vamos a proceder a dar de alta a una carrera:

Formulario Alta Carrera



Una vez añadida, podemos ver nuestra carrera insertada en la tabla:

Carrera	Distancia	Lugar	Fecha
Carrera 1	10 KM	Puerto del Rosario	19-06-2016
Carrera 1	5 KM	Puerto del Rosario	19-06-2016
Carrera 2	10 KM	Antigua	20-06-2016
Proyecto DAM	15 KM	Majada Marcial	19-06-2016

Modificar carreras:

Carreras
Añadir
Modificar
Eliminar
Generar documentos
Participantes
Inscripción
Modificar inscripción
Eliminar inscripción
Configuración
Usuarios
Cerrar Sesión

Formulario modificar Carrera

Seleccione el nombre de la carrera a modificar:

Nombre

- Seleccionar carrera -

Carrera	Distancia	Lugar	Fecha
Nombre Actual			
<input type="text"/>			
Nombre			
<input type="text"/>			
Lugar			
<input type="text"/>			
Fecha			
<input type="text"/>			
Distancias			
- Seleccionar distancia -	- Seleccionar distancia -	- Seleccionar distancia -	
<input type="button" value="+ refrescar"/>			
<input type="button" value="Modificar"/>			

En este apartado, podemos modificar los datos que queramos sobre una carrera en concreto. Este formulario, está compuesto por una lista desplegable donde seleccionaremos la carrera que queremos modificar, mostrándose en una tabla para visualizar los datos actuales y así saber que dato cambiar sin equivocaciones:

Formulario modificar Carrera

Seleccione el nombre de la carrera a modificar:

Nombre

Proyecto DAM

Carrera	Distancia	Lugar	Fecha
Proyecto DAM	15 KM	Majada Marcial	19-06-2016

una vez ahecho esto procedemos a realizar algún cambio, no es necesario rellenar todo el formulario ya que a lo mejor solo quieres modificar el nombre o la fecha. el campo que sí es necesario es el de nombre actual donde escribimos el nombre de la carrera que vamos a modificar:

Nombre Actual

Proyecto DAM

Nombre

Proyecto DAM modificado

Lugar

Majada Marcial modificado

Fecha

20-06-2016

Distancias

10 KM - Seleccionar distancia - Seleccionar distancia -

+ refrescar

Modificar

Antes:

Carrera	Distancia	Lugar	Fecha
Proyecto DAM	15 KM	Majada Marcial	19-06-2016

Después:

Carrera	Distancia	Lugar	Fecha
Proyecto DAM modificado	10 KM	Majada Marcial modificado	20-06-2016

Eliminar Carrera:

Carreras

Añadir

Modificar

Eliminar

Generar documentos

Participantes

Inscripción

Modificar inscripción

Formulario Eliminar Carrera

Carrera	Distancia	Lugar	Fecha
Carrera 1	10 KM	Puerto del Rosario	19-06-2016
Carrera 1	5 KM	Puerto del Rosario	19-06-2016
Carrera 2	10 KM	Antigua	20-06-2016
Proyecto DAM modificado	10 KM	Majada Marcial modificado	20-06-2016

Nombre

- Seleccionar nombre de usu - Eliminar

Aquí podemos eliminar las carreras simplemente seleccionándola en la lista desplegable y dando al botón eliminar:

Nombre

Proyecto DAM modificado ▼

Eliminar

Antes de eliminar la carrera nos saldrá un mensaje de confirmación:

localhost:8080 dice:

¿seguro que desea Borrar la carrera Proyecto DAM modificado?

Aceptar

Cancelar

Antes:

Carrera	Distancia	Lugar	Fecha
Carrera 1	10 KM	Puerto del Rosario	19-06-2016
Carrera 1	5 KM	Puerto del Rosario	19-06-2016
Carrera 2	10 KM	Antigua	20-06-2016
Proyecto DAM modificado	10 KM	Majada Marcial modificado	20-06-2016

Después:

Carrera	Distancia	Lugar	Fecha
Carrera 1	10 KM	Puerto del Rosario	19-06-2016
Carrera 1	5 KM	Puerto del Rosario	19-06-2016
Carrera 2	10 KM	Antigua	20-06-2016

Usuarios:

Carreras

Añadir

Modificar

Eliminar

Generar documentos

Participantes

Inscripción

Modificar inscripción

Nombre	Apellidos	Nombre de Usuario	Tipo de cuenta
pepe	hernandez perez	pepe	ADMIN

El apartado usuarios también se compone de Altas, Bajas y Modificaciones.

Insertar Usuario:

Este apartado nos permite insertar un usuario nuevo en la aplicación:

Formulario Alta Usuarios

Nombre

apellidos

Nombre de usuario

Contraseña

Vuelva a introducir la contraseña

Tipo de cuenta

- Seleccionar unidad -

Añadir

Para añadir el usuario debemos rellenar todos los campos del formulario, en caso de que alguno esté vacío, el sistema nos avisará, también dará aviso si las contraseñas no coinciden

Formulario Alta Usuarios

Nombre
Enrique

apellidos
San Blas García

Nombre de usuario
esbg

Contraseña

Vuelva a introducir la contraseña

Tipo de cuenta
Super Administrador

Añadir

Una vez insertado el usuario, lo podremos visualizar en pantalla:

Nombre	Apellidos	Nombre de Usuario	Tipo de cuenta
pepe	hernandez perez	pepe	ADMIN
Enrique	San Blas García	esbg	SUPERADMIN

Modificar Usuario:

Formulario Modificar usuario

Nombre de usuario a modificar:

Nombre

apellidos

Nombre de usuario

Contraseña

Vuelva a introducir la contraseña

Tipo de cuenta

- Seleccionar unidad -

Modificar

En este apartado podremos modificar cualquier campo de los usuarios que tenemos insertados.

Formulario Modificar usuario

Nombre de usuario a modificar:

esbg

Nombre

Juan

apellidos

García Díaz

Nombre de usuario

juan2

Contraseña

....

Vuelva a introducir la contraseña

....

Tipo de cuenta

Observador

Modificar

Nombre	Apellidos	Nombre de Usuario	Tipo de cuenta
pepe	hernandez perez	pepe	ADMIN
Juan	García Díaz	juan2	OBSERVADOR

Eliminar Usuario:

En este apartado tendremos que seleccionar un usuario de la lista desplegable y darle a eliminar:

Formulario Eliminar Usuario

Nombre

juan2

Eliminar

Nombre	Apellidos	Nombre de Usuario	Tipo de cuenta
pepe	hernandez perez	pepe	ADMIN