



# **Aplicación para la gestión de carreras**

**José Luis Escuer Toré**

Grado Multimedia – Trabajo fin de grado  
Área de Ingeniería Web

**Profesor colaborador - Ignasi Lorente Puchades**  
**Profesor responsable - Carlos Casado Martínez**

12-06-2017

# Licencia

## Memoria y presentación

La memoria y presentación del proyecto, cuando se haya realizado, estarán sujetos a la licencia que se indica a continuación:



Esta obra está sujeta a una licencia de [Reconocimiento-NoComercial-Compartirlgual 3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-sa/3.0/es/)

## La aplicación web tendrá copyright

© 2017 José Luis Escuer

Reservados todos los derechos. Está prohibida la reproducción total y parcial de esta obra a través de cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin autorización escrita de los autores o de los límites que autorice la Ley de Propiedad Intelectual.

## El Vídeo tendrá copyright

© 2017 José Luis Escuer Toré

Reservados todos los derechos. Está prohibida la reproducción total y parcial de esta obra a través de cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilm, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin autorización escrita de los autores o de los límites que autorice la Ley de Propiedad Intelectual.

## FICHA DEL TRABAJO FINAL

<b>Título del trabajo:</b>	<i>CronoDeportes</i>
<b>Nombre del autor:</b>	<i>José Luis Escuer Toré</i>
<b>Nombre del consultor/a:</b>	<i>Ignasi Lorente Puchades</i>
<b>Nombre del PRA:</b>	<i>Carlos Casado Martínez</i>
<b>Fecha de entrega (mm/aaaa):</b>	<i>06/2017</i>
<b>Titulación o programa:</b>	<i>Grado Multimedia</i>
<b>Área del Trabajo Final:</b>	<i>Ingeniería Web</i>
<b>Idioma del trabajo:</b>	<i>Castellano</i>
<b>Palabras clave:</b>	<i>Aplicación, Web, Multimedia</i>
<b>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de la aplicación, metodología, resultados y conclusiones del trabajo</i></b>	
<p>Cada vez hay más personas que practican deportes colectivos como ciclismo, running, etc. Esto está provocando cada vez se realizan más carreras por lo que se necesitan sistemas de control para que los participantes puedan ver el resultado así como su progreso y evolución.</p> <p>La finalidad de este TFG es desarrollar un sistema que permita registrar las distintas carreras con sus participantes, llevar el control de cada participante mostrando los datos en tiempo real para poder ser consultados, y poder ver los datos de los eventos que ya se hayan llevado a cabo.</p>	

**Abstract (in English, 250 words or less):**

More and more people are playing collective sports such as cycling, running, etc. This is causing more and more races so that control systems are needed in order to the participants can see the result as well as their progress and evolution.

The purpose of this TFG is to develop a system to record the different races with their participants, take control of each participant showing the data in real time to be consulted, and to see the data of events that have already been carried out.

# Índice

1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	2
1.3 Enfoque y método seguidos	3
1.4 Planificación del Trabajo	4
2. Análisis y diseño	7
2.1 Introducción	7
2.2 Contexto de uso	7
2.3 Requisitos	8
2.4 Casos de uso	9
2.5 Diagrama estático de análisis	17
2.6 Principios de Diseño	21
2.7 Diagrama de navegación	23
3. Conclusión	24
5. Bibliografía	25
6. Anexos	27

## Lista de figuras

Figura 1.- Participación Maratón de Barcelona	2
Figura 2.- Listado de tareas a realizar	5
Figura 3.- Diagrama de Gantt	6
Figura 4.- Diagrama de casos de uso	10
Figura 5.- Diagrama de clases	21
Figura 6.- Diagrama de navegación	23

# 1. Introducción

## 1.1 Contexto y justificación del Trabajo

En los países desarrollados, entre los cuales se encuentra nuestro país, cada vez las personas son más conscientes de la necesidad de cuidarse para que a medida que pasen los años se puedan enfrentar a una edad avanzada con garantías de estar en buenas condiciones físicas y psíquicas. Para ello, entre otras cosas, un número cada vez mayor de personas realizan ejercicio físico.

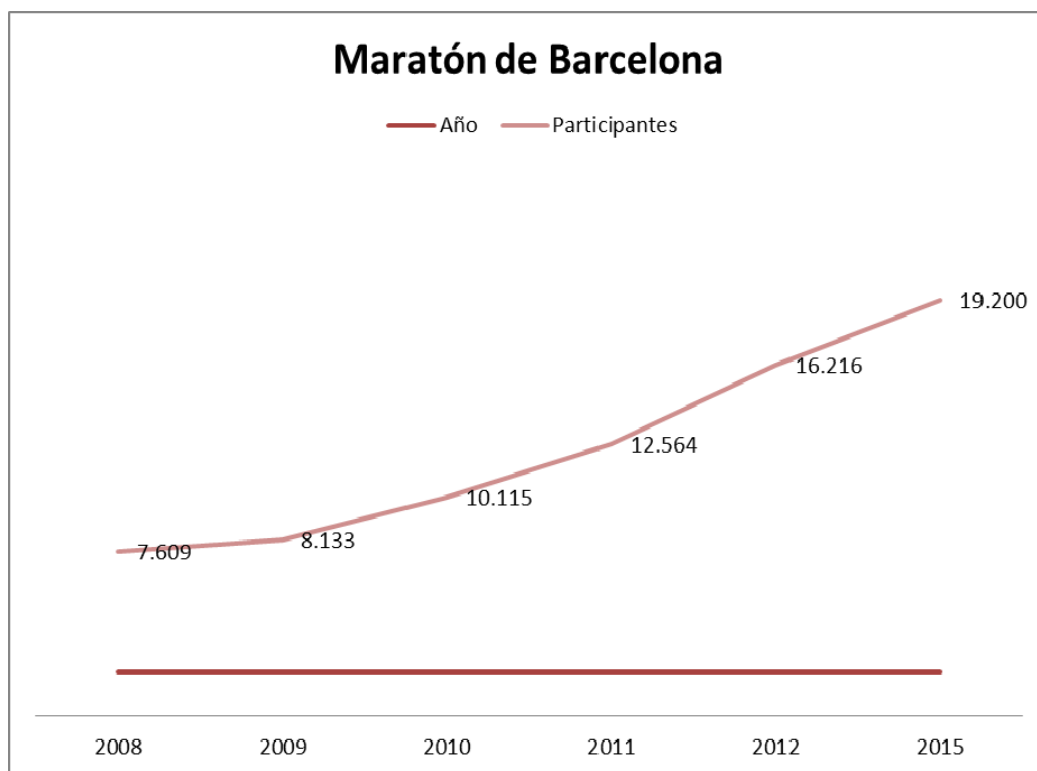
Cada vez hay más estudios en los que se demuestra el mayor número de beneficios para la salud al realizar una actividad física regular, de los que puede haber al llevar una vida sedentaria. Entre sus beneficios podemos mencionar que reduce la sensación de cansancio, que evita en mayor medida sufrir enfermedades cardiovasculares, o que ayuda a prevenir la obesidad beneficiando la movilidad articular, entre otros beneficios.

Otro efecto positivo es que se puede realizar junto a otras personas, y como somos seres que necesitamos la relación con otros seres de nuestra especie, es un aspecto beneficioso ya que de esta forma las personas se sienten que forman parte de un grupo, lo cual tiene beneficios psíquicos.

Desde que empezó la crisis económica en el año 2008, una gran parte de la población se ha visto afectada por recortes económicos o han recortado gastos por miedo a quedarse sin trabajo. Esto ha originado que gente que realizaba ejercicio en gimnasios o establecimientos similares haya sustituido este tipo de deportes por otros en los que no tenían que pagar una cuota mensual.

Entre los deportes que han experimentado mayor auge está el running debido principalmente a que la equipación necesaria para ejercer este deporte es relativamente barata, la crisis económica que ya hemos mencionado, y el contagio colectivo.

El efecto que tiene es que cada vez se realizan un mayor número de carreras, y cada vez hay más participantes, como ejemplo ponemos el siguiente gráfico en el que aparece la evolución de participantes en el tiempo en la Maratón de Barcelona.



*Figura 1.- Participantes Maratón de Barcelona en los últimos años*

En Ibiza, lugar en donde resido, este aumento ha sido muy notable debido a que se ha popularizado el triatlón, llegando a haber competiciones internacionales, lo que ha provocado un gran aumento de carreras de running, de ciclismo de montaña y carretera, y evidentemente de triatlón.

Por todo ello se ha planteado la necesidad de desarrollar un sistema para llevar la inscripción, el control, y estadística en este tipo de eventos o en otros que presenten las mismas características.

## 1.2 Objetivos del Trabajo

Los objetivos del trabajo son los siguientes:



- Como objetivo de la asignatura el primero sería el de realizar un proyecto completo partiendo desde cero, desde la idea hasta la finalización.
- Profundizar más en las distintas asignaturas que se necesitarán poner en práctica para el TFG, ya que hasta el momento, por la particularidad de la situación personal de un gran número de personas que estudian en la UOC, no hay tiempo de poner en práctica los conocimientos adquiridos en las distintas asignaturas una vez se han superado.
- Poner en práctica e ir aprendiendo más de los distintos lenguajes de programación que tendrán que usarse en conjunto, ya que se necesitará implementar casi todos los vistos hasta el momento para desarrollar las distintas partes de la aplicación web.
- Utilizar Arduino para la recepción de señales obtenidas mediante sensores, en este caso a través de un lector RFID, para la obtención, tratamiento, y visualización de los distintos datos que se vayan almacenando en la base de datos creada para ello.
- Poder afianzar lo aprendido para poder llevarlo a cabo en mi trabajo, con el objetivo de poder mejorar mi carrera profesional.

### **1.3 Enfoque y método seguido**

Se desarrollará una aplicación web para llevar el registro de carreras y participantes, y junto a ella se creará una maqueta a pequeña escala compuesta de un Arduino UNO con un módulo lector de RDIF que leerá los datos de las etiquetas dadas de alta en el sistema, obteniendo los tiempos de los que serían los distintos participantes que supuestamente se hayan inscrito en la competición. Los datos obtenidos se almacenarán en una base de datos mostrándolos en directo a través de la web, o consultándolos posteriormente a través del histórico.

Esta puede ser la mejor opción ya que hay lectores RFID de largo alcance, los cuales situados en distintos puntos de la carrera se pueden obtener tiempos

parciales y totales para carreras fuera de circuitos, y recoger los distintos tiempos de cada vuelta cuando la carrera se desarrolle en un circuito en el cual se tengan que dar más de una vuelta. Los distintos participantes llevarán dispositivos pasivos de bajo coste, pegatinas cuyo valor es de pocos céntimos de euro, las cuales tendrán un número no repetido para obtener los datos por cada uno de los participantes.

Para ello se usará la siguiente estrategia:

**Estudio:** de las tecnologías y dispositivos empleados, estudiando la viabilidad del producto en el caso de querer desarrollarlo para darle un uso real.

**Análisis:** de los datos recogidos en el estudio para concretar los requerimientos y recursos necesarios para desarrollar el producto.

**Diseño:** una vez finalizadas las fases anteriores se diseñará la arquitectura de la aplicación. Se concretarán los dispositivos necesarios que se usarán junto con el Arduino, el modelo de lector RFID, las tarjetas o pegatinas pasivas que usarán los participantes. También se diseñará toda la interfaz de la aplicación, la base de datos, etc.

**Implementación:** se llevará a cabo toda la fase de programación. Se irán probando las distintas partes que se vayan desarrollando para verificar que funcionan correctamente.

**Pruebas:** una vez se haya implementado se pasará a la fase de pruebas para verificar que todo funciona correctamente.

**Documentación:** se irá documentando las distintas fases del proyecto, algo muy importante para el posterior mantenimiento del producto.

## **1.4 Planificación del Trabajo**

El recurso que se va a emplear para desarrollar las tareas es el alumno que realiza este TFG.

A continuación se muestra el diagrama de Gantt en el que se especifican las tareas, la duración de ellas, y los hitos más relevantes, que son las distintas entregas que deben realizarse hasta la finalización.

Nombre	Fecha de inicio	Fecha de fin	Duración
☐ • TRABAJO FIN DE GRADO	22/02/17	12/06/17	74
☐ • PAC 1	22/02/17	7/03/17	9
• Lectura de materiales	22/02/17	2/03/17	6
• Búsqueda de informaci...	22/02/17	2/03/17	6
• Planificación	3/03/17	7/03/17	3
• Entrega PAC 1	7/03/17	7/03/17	0
☐ • PAC 2	8/03/17	5/04/17	21
• Búsqueda de informaci...	8/03/17	28/03/17	15
• Análisis	10/03/17	17/03/17	6
☐ • Diseño	20/03/17	5/04/17	13
• Arquitectura	20/03/17	23/03/17	4
• Diagramas UML	24/03/17	5/04/17	9
• Entrega PAC 2	5/04/17	5/04/17	0
☐ • PAC 3	6/04/17	12/05/17	23
☐ • Aprendizaje y repaso	6/04/17	5/05/17	18
• Bootstrap	6/04/17	12/04/17	5
• JQuery + Ajax	18/04/17	21/04/17	4
• MySQL + PHP	24/04/17	28/04/17	5
• Arduino	2/05/17	5/05/17	4
• Desarrollo	5/05/17	12/05/17	6
• Entrega PAC 3	12/05/17	12/05/17	0
☐ • Entrega Final	15/05/17	12/06/17	21
• Finalización memoria	15/05/17	24/05/17	8
• Realización video	25/05/17	7/06/17	10
• Presentación	8/06/17	12/06/17	3
☐ • Desarrollo	15/05/17	12/06/17	21
• Programación	15/05/17	12/06/17	21
• Seguridad	22/05/17	12/06/17	16
• Pruebas	22/05/17	12/06/17	16
• Fecha entrega Final	12/06/17	12/06/17	0

Figura 2.- Listado de tareas a realizar

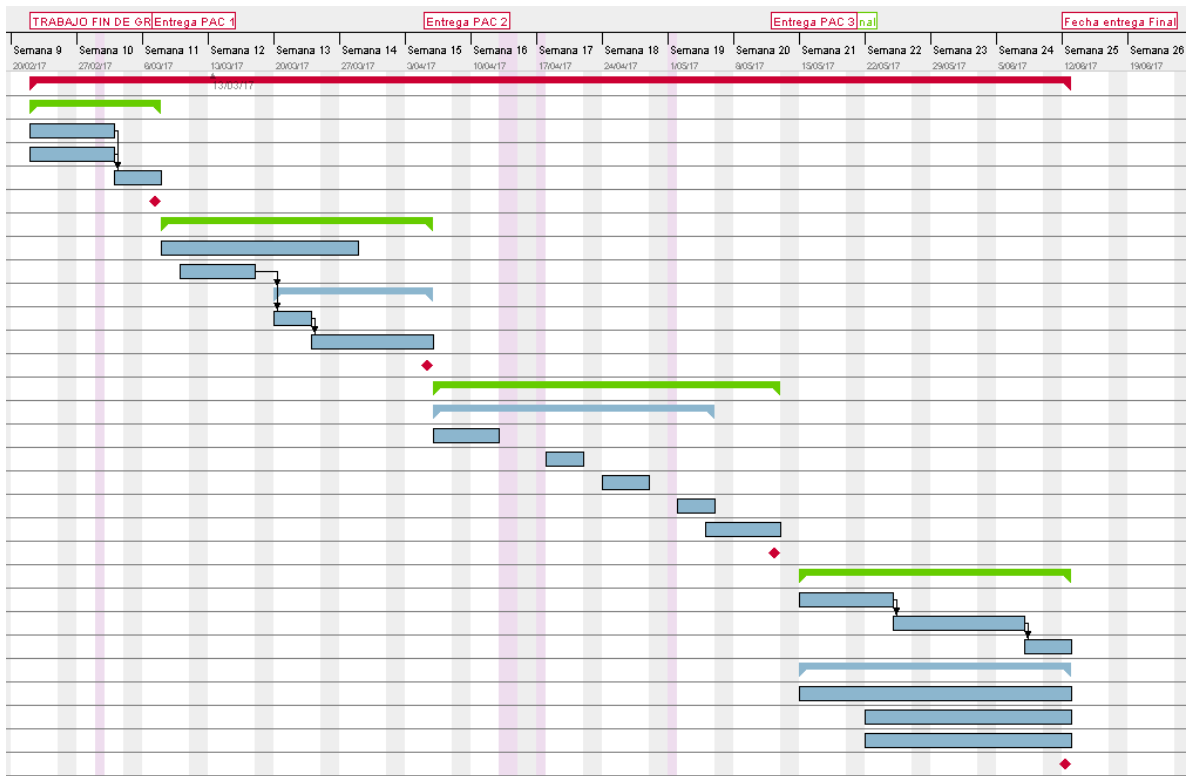


Figura 3.- Diagrama de Gantt

## 2. Análisis y diseño de la aplicación

### 2.1 Introducción

El análisis responde a la pregunta ¿Qué hace el programa? Y el diseño responde a la pregunta ¿Cómo lo hace?

El análisis y diseño consiste en la obtención de requisitos funcionales y no funcionales que va a tener la aplicación, para que el desarrollador del programa entienda las funciones requeridas, el comportamiento, el rendimiento y la interconexión.

Para que al final el programa funcione correctamente, es necesario tener una comprensión completa de los requisitos que va a tener. Es una tarea de investigación, descubrimiento y especificación.

Para ello se crean modelos de requisitos de datos, flujo de la información y control, así como del comportamiento operativo. Se analizan soluciones alternativas, se indica la interfaz con otros elementos y se establecen las restricciones.

### 2.2 Contexto de uso

Cada vez hay más gente que practica deporte al aire libre, y como las personas son seres sociables, se asocian con otras personas para practicar deporte en conjunto. Además de ser sociables las personas son también competitivas, por lo que les gusta de alguna manera poder medir la evolución que van teniendo fruto del entrenamiento habitual y constante, y les gusta poder compararla a la evolución que tienen otras personas que realizan actividades similares.

Por lo mencionado anteriormente se hacen carreras, y tienen cada vez más participación porque es la manera a través de la cual las personas pueden medir su evolución respecto a otras personas, a la vez que se establece un vínculo social con otras personas que tienen los mismos intereses e inquietudes.

Esa es la razón por lo que se crea éste sistema, para obtener un sistema fiable y preciso mediante el que se pueda establecer los tiempos de estas personas, y a su vez mantener un histórico de datos a través del cual puedan medir su evolución.

## 2.3 Requisitos

En este apartado se tiene que analizar los requisitos que tendrá la aplicación. Para ello distinguimos los requisitos funcionales, que responden a la pregunta ¿Qué hace la aplicación? Y los requisitos no funcionales, que responden a la pregunta ¿Cómo lo hace la aplicación?

En esta parte normalmente se recogen los requisitos que se recogen del cliente que ha hecho el encargo, y este no es el caso, ya que no hay alguien por detrás que nos haya encargado el producto y nos haya puesto una serie de requisitos y restricciones. Pero, de todas formas, se establecerán unos requisitos básicos que la aplicación tendrá que cumplir.

### Requisitos funcionales

En este apartado vamos a identificar algún ejemplo de requisitos funcionales que va a tener que cumplir el sistema.

- El corredor inscrito en una carrera querrá poder comprobar los resultados de la carrera una vez haya terminado.
- Los usuarios registrados en una carrera deberían de poder ponerse en contacto con los gestores del portal para poder expresar alguna duda, queja o sugerencia.
- Además de ver la información de otros servicios que se presten, cualquier persona tendría que poder pedir más información por medio electrónico.

### Requisitos no funcionales

En este apartado vamos a identificar algún ejemplo de requisitos no funcionales que va a cumplir la aplicación.

- La navegación será fácil y sencilla para que la aplicación pueda ser usada por gente con pocos conocimientos de informática. Para verificar que se cumple este requisito se harán pruebas por personas que cumplan ese perfil.

- La interfaz será sencilla y a la vez vistosa para que la gente se sienta atraída. Se harán pruebas con distintos usuarios para verificar que la interfaz les resulta atractiva.
- El usuario organizador será el único tipo de usuario que podrá dar de alta carreras. Se harán pruebas para verificar que el organizador es el único tipo de usuario que puede hacerlo.
- La inscripción a una carrera sólo podrá ser hecha por el usuario que se ha validado en la página y que tiene un perfil de corredor. Se probará que un usuario no registrado u otro perfil de usuario no se pueden inscribir en una carrera.

## 2.4 Casos de uso

En este apartado se van a describir los casos de uso que tendrá la aplicación, es decir, la secuencia que va a seguir un usuario para llevar a cabo una acción teniendo en cuenta el flujo de información alternativo.

El sistema tendrá tres actores, que son el Usuario, que es la persona que accede a la página sin haber hecho el Login; después está el actor Corredor, que es el usuario que está registrado en la página como tal, principalmente para poder inscribirse en alguna. Por último está el actor Organizador, que es el usuario que está registrado en la página principalmente para poder dar de alta alguna carrera. Más adelante se describen y se indican los casos de uso que van a poder realizar cada uno de los actores.

El caso de uso “Registrarse” extiende el caso de uso “Login”. Esto significa que si el usuario se registra podrá hacer el caso de uso “Login” en ese momento, pero el caso de uso “Login” se puede usar de manera independiente de aquél.

Los casos de uso “Dar de alta carrera”, “Inscribirse en carrera, incluyen el caso de uso “Login”, ya que para llegar a alguno de esos casos se habrá tenido que validar el usuario.

Los casos de uso “Inscribirse en carrera”, y “Ver datos de carreras” incluyen el caso de uso “Seleccionar carrera”, ya que antes de llegar allí el usuario habrá tenido que seleccionar una carrera.

El diagrama de casos de uso permite visualizar de forma rápida la funcionalidad que tendrá cada actor, ya que contiene los actores y los casos de uso, y en este caso quedaría de la siguiente forma:

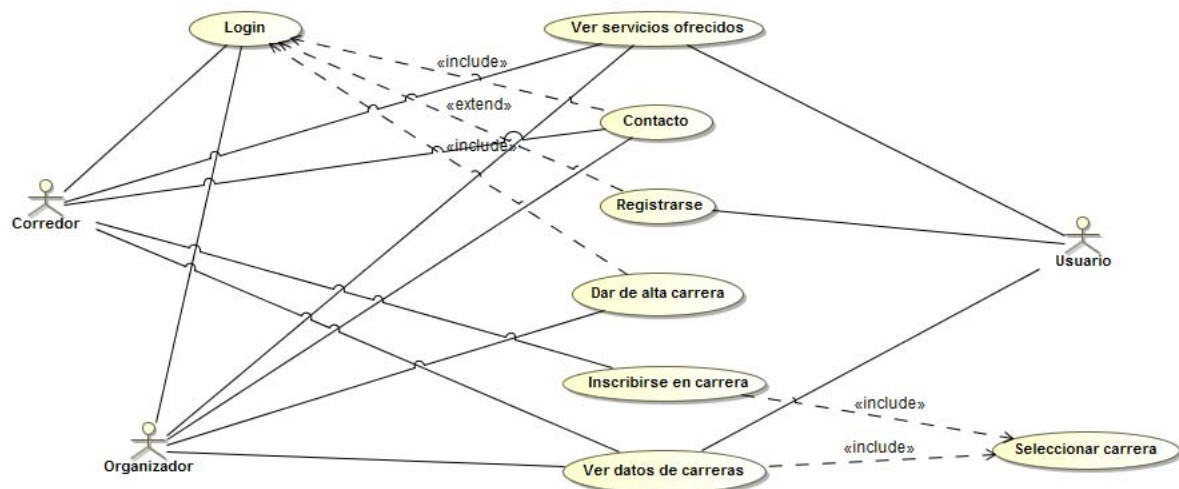


Figura 4.- Diagrama de casos de uso

Además del diagrama, vamos a hacer la especificación de los casos de uso de forma textual, de tal forma que nos permita tener una idea de cómo va a ser la interfaz de la aplicación.

### Caso de uso “Ver servicios ofrecidos”

Nombre: Ver servicios ofrecidos

Actores: lo pueden usar el usuario, el corredor y el organizador.

Resumen de la funcionalidad: permite ver los servicios que se ofrecen.

Flujo de acontecimientos principal:

- 1) El usuario está en la ventana en donde se muestran los servicios ofrecidos.

Flujo de acontecimientos alternativos:



Pantallas:

Servicios

- Lista de los servicios mostrando una imagen representativa del servicio y una breve descripción de cada uno

### Caso de uso “Contacto”

Nombre: Contacto

Actores: lo pueden usar el usuario con perfil de corredor y el usuario con perfil de organizador.

Resumen de la funcionalidad: permite contactar con los gestores de la página.

Flujo de acontecimientos principal:

- 1) El usuario accede al menú de navegación Contacto.
- 2) Se muestran los datos del formulario.
- 3) El usuario rellena el formulario, todos los campos son obligatorios, y selecciona el botón “Enviar”.
- 4) Se comprueba que los datos sean correctos.
- 5) Se insertan los datos en la base de datos.

Flujo de acontecimientos alternativos:

- 3a) El usuario cancela la petición de información.
  - 3a1) Finaliza el caso.
- 4a) El sistema indica el error de los datos introducidos.
  - 4a1) Se vuelve a mostrar la ventana del formulario mostrando el error.
  - 4a2) Pasamos al punto 3.

Pantallas:

Contacto

- El sistema muestra el formulario de contacto junto con el botón “Enviar”.

### Caso de uso “Registrarse”

Nombre: Registrarse

Actor: lo puede usar el usuario.

Resumen de la funcionalidad: permite al usuario registrarse en el sistema para poder usar otras funcionalidades.

Casos de uso relacionados: extiende el caso de uso "Login", ya que se podrá usar después de registrarse o el usuario podrá decidir no hacerlo.

Flujo de acontecimientos principal:

- 1) Se muestra el formulario con los datos del registro.
- 2) El usuario decide si va a registrarse con el perfil de corredor o con el perfil de organizador.
- 3) El usuario introduce los datos y selecciona el botón "Aceptar".
- 4) Se comprueba que los datos introducidos son correctos.
- 5) El sistema envía un correo electrónico al usuario con los datos del registro.

Flujo de acontecimientos alternativos:

- 2a) El usuario cancela la operación.
  - 2a1) El caso de uso termina.
  - 3a) El sistema indica el error de los datos introducidos.
    - 3a1) Se vuelve a mostrar la ventana de registro mostrando el error.
    - 3a2) Pasamos al punto 2.

Pantallas:

Registro

- Se muestran los datos que se solicitan para poder registrarse: correo, contraseña, repetir la contraseña, nombre, apellidos, provincia, tipo de usuario. Una vez se haya elegido el tipo de usuario:
  - o Si tendrá un perfil corredor aparecen los campos de fecha de nacimiento, DNI, sexo, la talla de camiseta, y el botón de aceptar.
  - o Si tendrá un perfil organizador aparecen los campos empresa, el CIF y el botón de aceptar.

## Caso de uso “Dar de alta carrera”

Nombre: Dar de alta carrera

Actor: lo puede usar el usuario con perfil de organizador.

Resumen de la funcionalidad: permite dar de alta una nueva carrera.

Flujo de acontecimientos principal:

- 1) El usuario entra en la página (caso de uso incluido “Login”).
- 2) El sistema le redirige al apartado de perfil de usuario organizador.
- 3) Selecciona el botón “Crear carrera” para crear la nueva carrera.
- 4) El usuario introduce los datos y selecciona el botón “Crear carrera”.
- 5) Se comprueba que los datos introducidos son correctos.

Flujo de acontecimientos alternativos:

- 4a) El usuario cancela la operación.
  - 4a1) El caso de uso termina.
- 5a) El sistema indica el error de los datos introducidos.
  - 5a1) Se vuelve a mostrar la ventana del formulario mostrando el error.
  - 5a2) Pasamos al punto 4.

Pantallas:

Menú usuario

- Se muestra las opciones del menú de usuario organizador, y un botón de “Crear carrera”.

Formulario

- Se muestra la ventana con el formulario para introducir los datos, así como el botón de “Crear carrera” para aceptar los datos introducidos.

## Caso de uso “Inscribirse en carrera”

Nombre: Inscribirse en carrera

Actor: lo puede usar el corredor.

Resumen de la funcionalidad: permite al usuario inscribirse en una carrera.

Flujo de acontecimientos principal:

- 1) El usuario entra en la página (caso de uso incluido "Login")
- 2) El sistema le redirige al apartado de perfil de usuario corredor.
- 3) Se dirige al apartado de carreras
- 4) El usuario selecciona una carrera (caso de uso incluido "Seleccionar carrera").
- 5) El usuario selecciona el botón "Inscribirse".
- 6) El sistema registra la inscripción
- 7) El caso de uso finaliza.

Pantallas:

- Ventana con los datos de la carrera, un botón para inscribirse en la carrera y otro botón para volver a la ventana anterior.
  - o Si la inscripción se realiza sale un mensaje indicándolo.
  - o Si el corredor ya está inscrito sale un mensaje indicándolo, y no se realiza la inscripción.
  - o Si se ha llegado al número máximo de participantes sale un mensaje indicándolo, y no se realiza la inscripción.

### **Caso de uso "Ver datos de carreras"**

Nombre: Ver datos de carreras

Actores: lo pueden usar el usuario, el corredor y el organizador.

Resumen de la funcionalidad: permite ver los datos de las carreras, obteniendo los resultados.

Flujo de acontecimientos principal:

- 1) El usuario selecciona una carrera (caso de uso incluido "Seleccionar carrera").
- 2) El sistema muestra el resultado de los corredores inscritos en esa carrera.

- 3) El usuario selecciona "Volver".
- 4) El caso de uso finaliza.

Pantallas:

Clasificaciones

- Listado de participantes ordenado por la clasificación de los corredores inscritos en esa carrera que han participado, si es el momento en el que se está desarrollando una carrera se ve cómo se va actualizando la clasificación según van pasando los participantes por los distintos puntos de control que puede haber. Botón "Volver" para volver al listado de clasificaciones.

### Caso de uso "Login"

Nombre: Login

Actores: lo pueden usar el corredor y el organizador.

Resumen de la funcionalidad: permitir que el usuario se pueda identificar para entrar en la página y poder acceder a las opciones del menú de usuario.

Casos de uso relacionados: tal y como hemos dicho, este caso de uso está incluido en los casos de uso "Dar de alta carrera", "Inscribirse en carrera" y "Contacto".

Flujo de acontecimientos principal:

- 1) Se muestra la ventana de "Login".
- 2) El usuario introduce sus datos que le permitirán acceder al sistema.
- 3) Se comprueba que los datos sean correctos.

Flujo de acontecimientos alternativos:

- 2a) El usuario le da a "Cancelar".
  - 2a1) Finaliza el caso.
- 3a) El sistema indica el error de los datos introducidos.
  - 3a1) Se vuelve a mostrar la ventana de login mostrando el error.
  - 3a2) Pasamos al punto 2.

Pantallas:

Login

- Los datos de usuario y contraseña, y el botón "Entrar". Debajo un enlace en donde un usuario se puede registrar en el sistema.

Menú usuario

- En la parte izquierda un menú con las opciones del menú del usuario, que cambiarán si el usuario ha entrado como corredor o ha entrado como organizador. A la derecha aparece un formulario en donde se podrían modificar los datos del usuario.

### Caso de uso "Seleccionar carrera"

Nombre: Seleccionar carrera

Actores: lo pueden usar el usuario, el corredor y el organizador.

Resumen de la funcionalidad: seleccionar una carrera de las que se muestren en el listado de carreras

Este caso de uso está incluido en los casos de uso "Inscribirse en carrera", y "Ver datos de carrera".

Flujo de acontecimientos principal:

- 1) El sistema muestra una lista de carreras.
- 2) El usuario selecciona una carrera.
- 3) El sistema muestra la ventana con la descripción de la carrera.

Pantallas:

Carreras

- Se muestra la lista de carreras. Se muestra una imagen representativa de la carrera junto con una breve descripción.

### **Carrera que no se ha realizado**

- Si el usuario ha entrado como corredor, se muestra una imagen de la carrera junto con toda su información, y habrá un botón “Inscribirse” para poder inscribirse en la carrera, junto a otro para volver. Si el usuario ha entrado como organizador o es un usuario que no ha hecho login el sistema muestra toda la información de la misma, el botón de volver, el botón de inscribirse deshabilitado, y debajo un mensaje indicando que sólo se puede inscribir un usuario con perfil corredor.  
Debajo se muestra el mapa indicando el punto en donde se va a realizar la carrera.

### **Carrera que ya se ha realizado**

- En este caso todos los usuarios que hayan hecho login y los que no lo hayan hecho verán lo mismo. Se muestra una imagen de la carrera junto con toda su información y un botón de volver.  
Debajo aparece una tabla con la clasificación de la carrera ordenada por la posición en la que han quedado los participantes.  
Debajo se muestra el mapa indicando el punto en donde se va a realizar la carrera.

## **2.5 Diagrama estático de análisis**

En este diagrama se representa lo que conoce el sistema del mundo real desde el punto de vista del análisis, por lo que no se tiene en cuenta la tecnología que se va a utilizar. La representación se hace mediante un diagrama de clases UML, en el que se pone una clase por cada elemento del mundo real que se usa junto con sus atributos, y las relaciones que se dan entre las distintas clases.

Lo primero que hay que hacer es identificar las clases y los atributos de los objetos. En este apartado vamos a obtener todas las abstracciones básicas del sistema.

### **Clases, atributos y relaciones**

Las clases obtenidas a través del análisis realizado son

**Usuario:** son las personas que se han dado de alta en el sistema. Las personas que quieren inscribirse en alguna carrera, organizar carreras, o realizar otra serie de cosas. Hay alguna funcionalidad que la podrá utilizar un usuario sin estar registrado, pero en ese caso no será un objeto de esta clase.

Los atributos de la clase que se necesitan conocer son los siguientes:

- correo: este atributo servirá para hacer el “Login” en el sistema.
- contraseña: servirá para poder validarse y entrar en la página.
- nombre: es el nombre de la persona que entra, para poder referirse a ella cuando se saquen listados o se hagan otro tipo de cosas.
- apellidos: igual que el anterior.
- tipo: en donde se guarda el tipo de usuario para saber el perfil que va a tener.

**Corredor:** es el usuario que se ha dado de alta en el sistema con la intención principal de poderse inscribir en alguna de las carreras que se van a producir. Está relacionado con la clase usuario, ya que aquella contendrá parte de sus datos.

Además de los atributos de la clase usuario, también necesitará conocer los atributos siguientes:

- fecha: es un atributo que servirá para guardar la fecha de nacimiento del corredor con los objetivos de poder sacar estadísticas, así como también el de poder hacer clasificaciones por edades.
- DNI: es el atributo que servirá para poder identificar al corredor.
- sexo: al igual que el atributo fecha, se guardará con los objetivos de poder sacar estadísticas, así como también el de poder hacer clasificaciones por sexo..
- talla: servirá para que al inscribirse en una carrera se le pueda asignar la talla de la camiseta en función del dato que haya introducido.

**Organizador:** es el usuario que se ha dado de alta en el sistema con la intención principal de poder crear carreras. Normalmente suele ser una empresa que se dedica a organizar este tipo de eventos. Está relacionado con la clase usuario, ya que aquella contendrá parte de sus datos. Cada organizador podrá ser una



agregación de cualquier número de carreras, de tal forma que cada carrera tendrá que conocer quién será el organizador de aquella.

Los atributos de la clase que se necesitan conocer son los siguientes:

- empresa: como ya se ha mencionado, el nombre de la empresa que crea la carrera, o en su defecto el nombre de la persona que la va a organizar.
- CIF: es el número de identificación fiscal de la empresa, o en su defecto el DNI de la persona organizadora.

**Carrera:** es la clase en donde se guardarán todos los datos de la carrera.

Los atributos de la clase que se necesitan conocer son los siguientes:

- nombre: es el nombre que se le va a dar a la carrera.
- tipo: se necesita conocer el tipo de la carrera para que los corredores que se quieran inscribir conozcan a qué tipo pertenece.
- fecha: se necesita conocer la fecha de celebración de la carrera.
- lugar: el sitio donde se celebrará (podrá ser el nombre de un pabellón, etc.).
- distancia: corresponde a la longitud total del recorrido de la carrera.
- longitud y latitud: son las coordenadas para poder ubicar el sitio en un mapa.
- puntos: necesitamos conocer los puntos de control que va a tener cada carrera para poder reflejar los resultados al mostrar los resultados.
- imagen: de cada carrera se podrá guardar una imagen representativa del evento.
- participantes: se necesitará saber el número total de participantes en una carrera para no superar el número máximo cuando los corredores se inscriban en ella.

**Inscrito:** corresponde a la clase que contendrá los corredores inscritos en una carrera, siendo una asociación de ambas clases.

Los atributos de la clase que se necesitan conocer son los siguientes:

- dorsal: se necesitará conocer el dorsal de cada corredor inscrito en la carrera.
- punto: se necesitará conocer el punto de la carrera en el que se tomará el tiempo.
- tiempo: se necesitará conocer el tiempo obtenido en ese punto de la carrera.

**Servicio:** se publicitan los distintos servicios que se ofrecen desde la página.

Los atributos de la clase que se necesitan conocer son los siguientes:

- nombre: se necesitará conocer el nombre del servicio.
- descripción: para poder conocer más datos acerca del servicio.

**Petición:** de esta clase se obtendrán todas las peticiones que se hagan respecto a los servicios ofrecidos. Cada servicio será una agregación de cualquier número de peticiones, de tal forma que cada petición tendrá que conocer el servicio al que se refiere la petición, pero el servicio no conocerá las peticiones que habrá sobre él. Como excepción tendremos las peticiones que se hagan en el apartado “contacto” de la aplicación, ya que estas se guardarán aquí con un identificador que no corresponderá a ninguno de los servicios ofrecidos.

Los atributos de la clase que se necesitan conocer son los siguientes:

- fecha: se necesitará conocer la fecha en la que se ha realizado la petición por parte del usuario registrado.
- teléfono: aunque cada vez es más habitual el contacto por correo electrónico, se necesitará conocer el teléfono para poder contactar por esta vía de forma extraordinaria.
- msg: se necesitará conocer el contenido de la petición.

**Provincia:** de esta clase se obtendrán todas las provincias. Se utilizarán al registrarse los usuarios y cuando al dar de alta una carrera.

- provincia: contendrá el nombre de la provincia.

**Tag:** en esta clase se guardarán los números de etiquetas RFID. Cada una de ellas se asociará a un usuario con perfil corredor al inscribirse a una carrera.

- tag: contendrá el número de etiqueta que está formado por doce números hexadecimales.

Una vez hemos definido las distintas clases con sus atributos, ya podemos establecer el diagrama en el cual se van a especificar las relaciones entre las distintas clases, cuando las haya, sus atributos, la cardinalidad, y la

navegabilidad. De esta forma podemos ver gráficamente los conceptos del dominio que se está desarrollando.

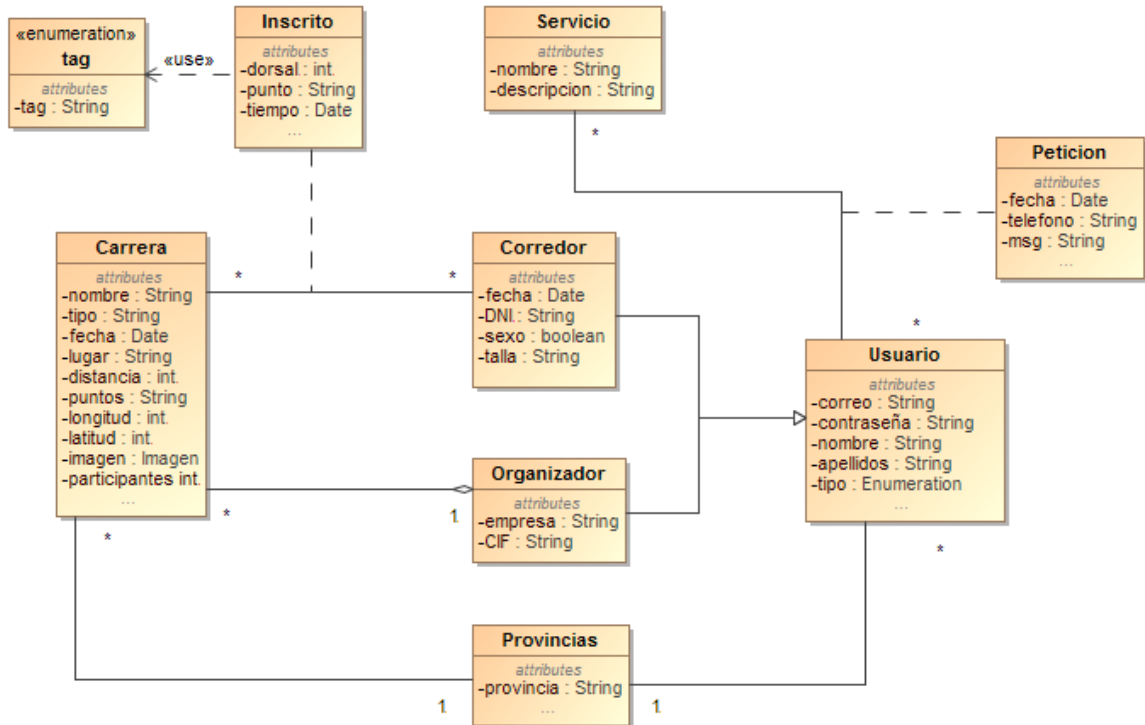


Figura 5.- Diagrama de clases

## 2.6 Principios de diseño

Para mejorar la calidad del desarrollo se usarán los principios de diseño, que consiste en aplicar una serie de reglas para que el desarrollo de la programación orientada a objetos sea más robusto, consiguiendo un software más fácil de entender, mantener, y reutilizar.

Principalmente se usarán estos principios:

### Bajo acoplamiento

Consiste en hacer que una clase no dependa de otra. Esto se consigue, por ejemplo, no accediendo desde la clase "A" a los atributos de la clase "B" directamente, sino a través de su interfaz, de tal forma que si tenemos que modificar los atributos de la clase "B", o alguna de sus operaciones, mientras no modifiquemos la interfaz no tendremos que preocuparnos de modificar el

contenido de la clase “A”, o de otras clases que accedan a los atributos de la misma forma.

### Alta cohesión

Con este principio conseguimos hacer una clase lo más sencilla posible. Se trata de hacer que la clase tenga las operaciones estrictamente necesarias que se vayan a usar en otros sitios y no más, y que si queremos añadir alguna funcionalidad en vez de ampliar la clase con nuevas operaciones creamos otras clases para dotarlas de esa responsabilidad. Llevando al extremo ese principio tenemos el principio de responsabilidad única, que consiste en hacer que cada clase tenga una única responsabilidad, es decir, que se simplifica al máximo el contenido de una clase.

## 2.7 Diagrama de navegación

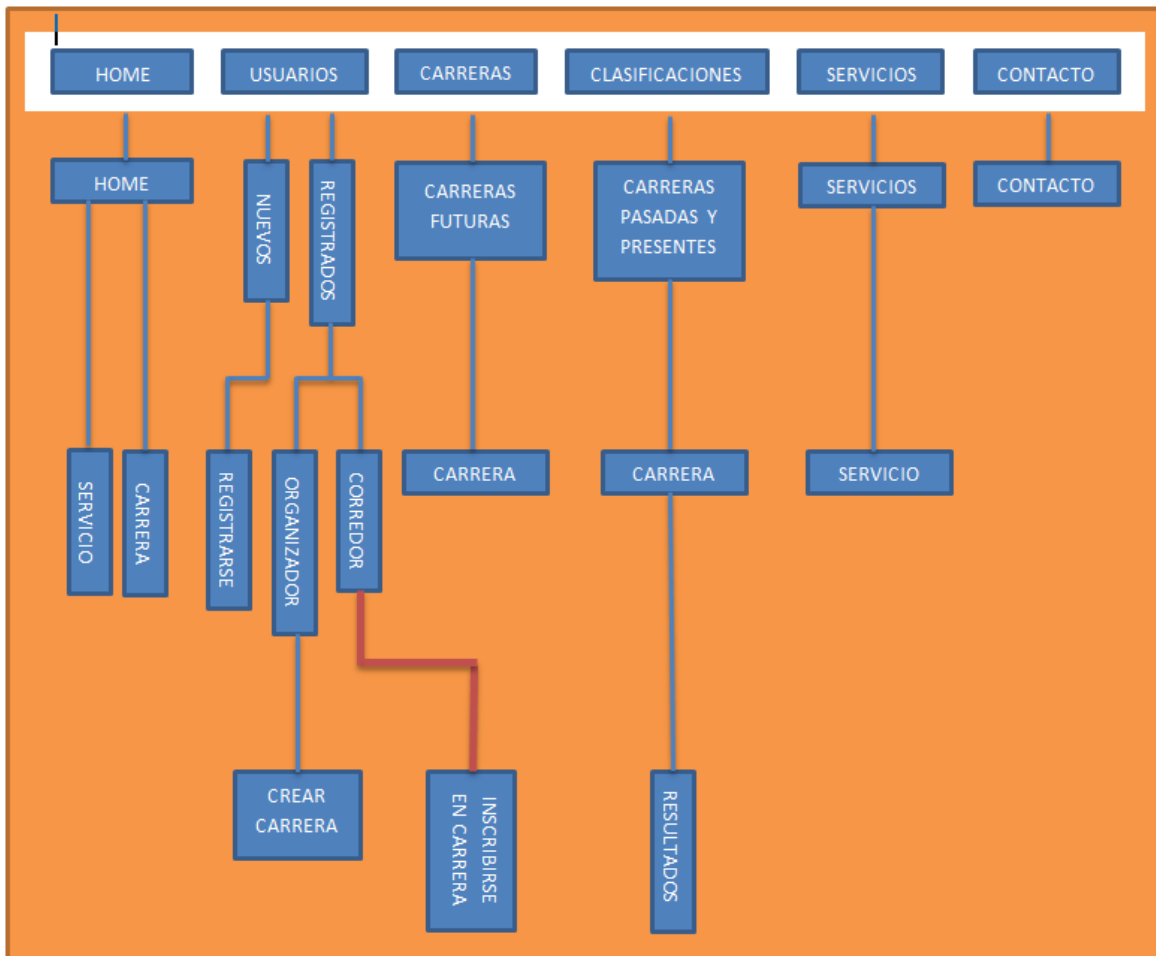


Figura 6 – Diagrama de navegación

## 3. Conclusiones

### Conclusiones del trabajo

Las conclusiones que podemos sacar, como resultado del trabajo realizado son:

- Al ser un trabajo con diversas fases, es necesario realizar una buena planificación.
- Siempre hay que ser pesimista en la estimación de los tiempos, ya que a lo largo del tiempo van saliendo imprevistos, tanto internos del propio trabajo como externos del entorno que nos rodea, que provocan la dilatación en el tiempo de las distintas fases, y este retraso se va acumulando para la planificación y realización de las siguientes.

Los objetivos planteados al inicio del trabajo se han cumplido, puesto que eran unos objetivos generales que se iban a lograr si se realizaba el proyecto.

- Realización del proyecto completo
- Profundizar en las asignaturas del grado, ya que reflexionando se han aplicado los conocimientos adquiridos en muchas de las asignaturas, en unos casos de manera directa, lo aprendido en las asignaturas de programación, y en otros casos de manera indirecta, asignaturas más de diseño, usabilidad, narrativa, etc.

La planificación se ha ido modificando en cada entrega, ya que como he dicho al principio hay factores internos y externos que han afectado a la planificación. Ha habido algún apartado que se quería realizar al principio y que se ha tenido que modificar o suprimir para poder llegar a la finalización del mismo. No obstante, las funcionalidades principales del proyecto sí que se han realizado.

En un futuro se podrían añadir funcionalidades al proyecto, o incluso convertirlo en realidad adaptando las herramientas empleadas, como lectores rfid, etc. Por unas profesionales de largo alcance.

## 4. Bibliografía

- MDG-Carreras. (2017). *Equipos de cronometraje para eventos deportivos: Atletismo/Triatlón/Motocross/Bike*. [online] Available at: <http://www.mdgcarreras.com/> [Accessed 5 Apr. 2017].
- ThemeForest. (2017). *Website Templates & Themes from ThemeForest*. [online] Available at: <https://themeforest.net/> [Accessed 5 Apr. 2017].
- Raya Martos, Jose y Pradel Miquel, Jordi (2015). *Introducció a l'enginyeria del programari*. Barcelona: FUOC
- Raya Martos, Jose y Pradel Miquel, Jordi (2015). *Orientació a objectes*. Barcelona: FUOC
- Raya Martos, Jose y Pradel Miquel, Jordi (2015). *Requisits*. Barcelona: FUOC
- Raya Martos, Jose y Pradel Miquel, Jordi (2015). *Anàlisi UML*. Barcelona: FUOC
- Raya Martos, Jose y Pradel Miquel, Jordi (2016). *Caso práctico de aplicación de patrones*. Barcelona: FUOC
- Raya Martos, Jose y Pradel Miquel, Jordi (2016). *Introducción a los patrones*. Barcelona: FUOC
- Raya Martos, Jose y Pradel Miquel, Jordi (2016). *Catálogo de patrones*. Barcelona: FUOC
- Ardumania.es. (2017). *Aprende | Curso iniciación | Arduino | Ardumanía*. [online] Available at: <http://www.ardumania.es/aprende/> [Accessed 02 May 2017].
- bldr. (2017). *bldr » Can You ID This? ID-20 RFID Reader + Arduino*. [online] Available at: <http://bldr.org/2011/02/rfid-arduino/> [Accessed 03 May 2017].
- Arduino.cc. (2017). *Arduino - Home*. [online] Available at: <https://www.arduino.cc/> [Accessed 05 May 2017].
- Prometec.net. (2017). *Primeros pasos en Arduino | Tutoriales Arduino*. [online] Available at: <http://www.prometec.net/modulo-1/> [Accessed 04 May 2017].
- Tutorial by Cytron. (2017). *RFID Reader ID-20LA (125kHz) with Arduino - Tutorial by Cytron*. [online] Available at: <http://tutorial.cytron.com.my/2015/02/17/rfid-reader-id-20la-125khz-with-arduino/> [Accessed 10 May 2017].
- YouTube. (2017). *Tu propio Servidor IoT - Cap #5*. [online] Available at: <https://www.youtube.com/watch?v=zQY6Pfpw8> [Accessed 10 May 2017].
- YouTube. (2017). *Curso completo de Bootstrap desde cero 1.- Introducción e Instalación*. [online] Available at: [https://www.youtube.com/watch?v=nug1pMke-y4&index=1&list=PLhSj3UTs2\\_yWTKvu1Aq3xUhzIjNBZ3MFW](https://www.youtube.com/watch?v=nug1pMke-y4&index=1&list=PLhSj3UTs2_yWTKvu1Aq3xUhzIjNBZ3MFW) [Accessed 06 Abril 2017].
- YouTube. (2017). *Curso jQuery - Introducción*. [online] Available at: <https://www.youtube.com/watch?v=jKbjblt4NXA&list=PLTE0X123tz-yxigJHnmuEgqWeVMI7YI-W> [Accessed 18 Abril 2017].

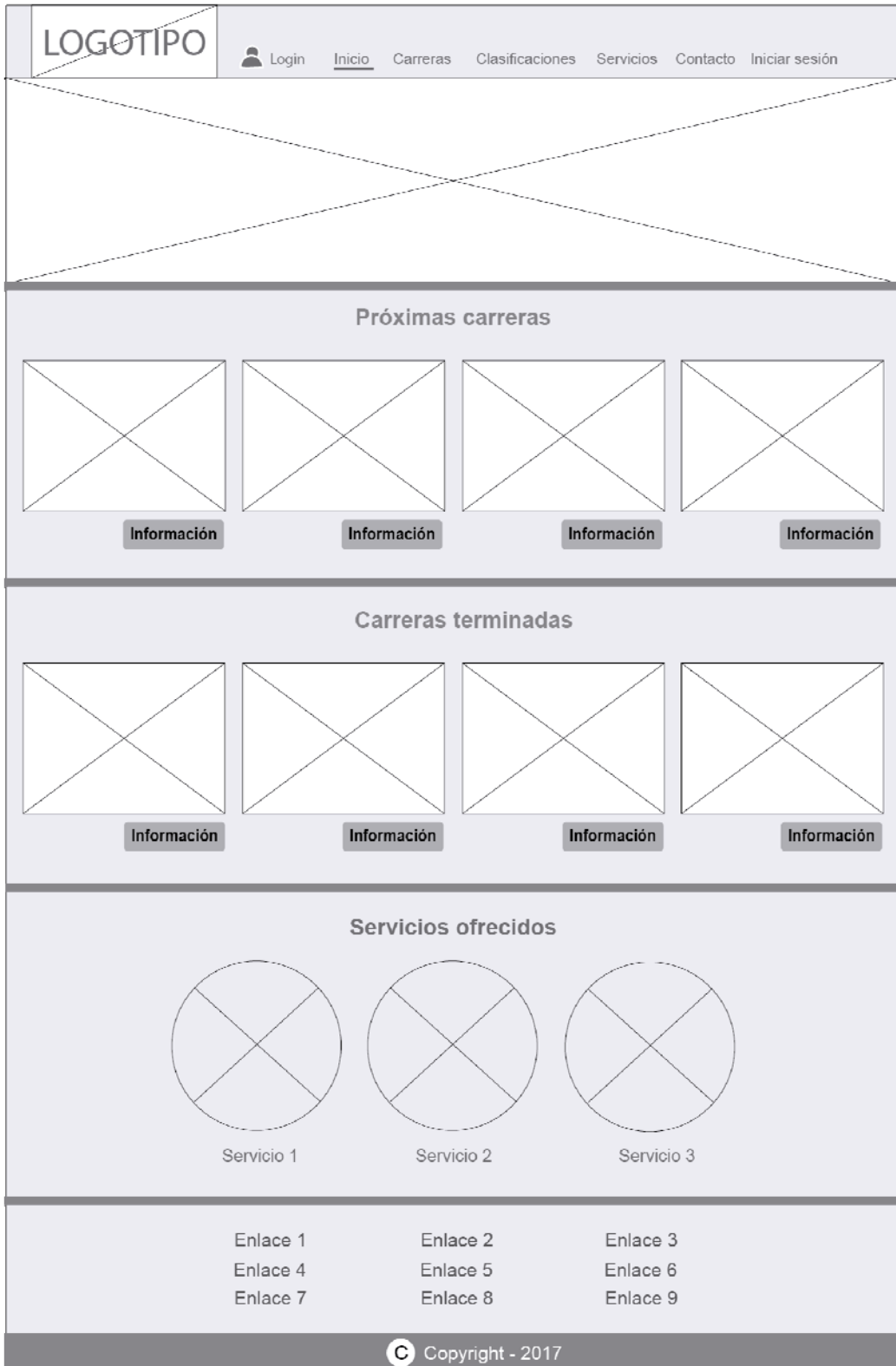
Udemy. (2017). *PHP y MYSQL: El Curso Completo, Practico y Desde 0 !*. [online] Available at: <https://www.udemy.com/php-y-mysql/learn/v4/t/lecture/4341366?start=0> [Accessed 24 Abril 2017].




# 5. Anexos

## Anexo 1.- Prototipos a baja resolución

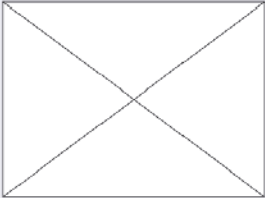
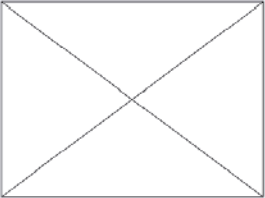
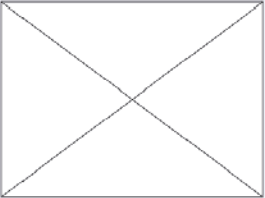
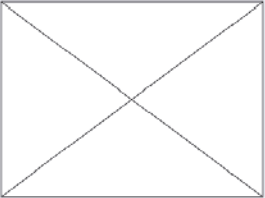
### Página home



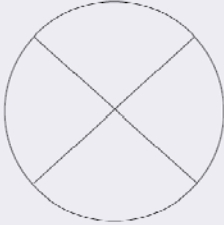
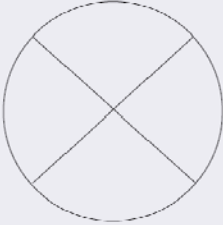
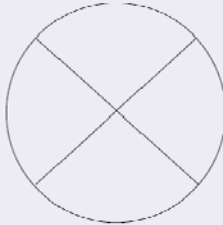
LOGOTIPO  Login Inicio Carreras Clasificaciones Servicios Contacto Iniciar sesión

---

### Próximas carreras

			
Información	Información	Información	Información

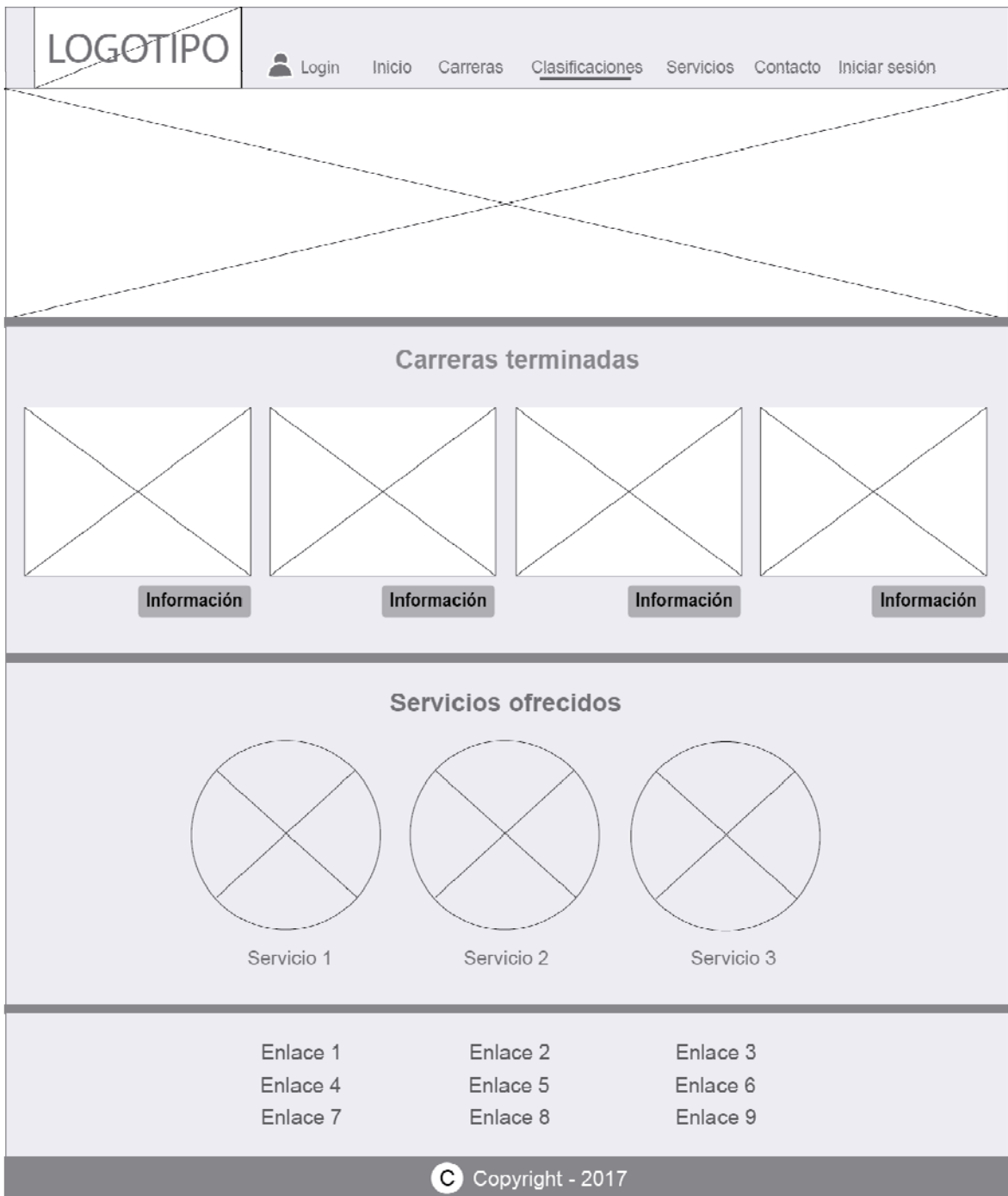
### Servicios ofrecidos

		
Servicio 1	Servicio 2	Servicio 3

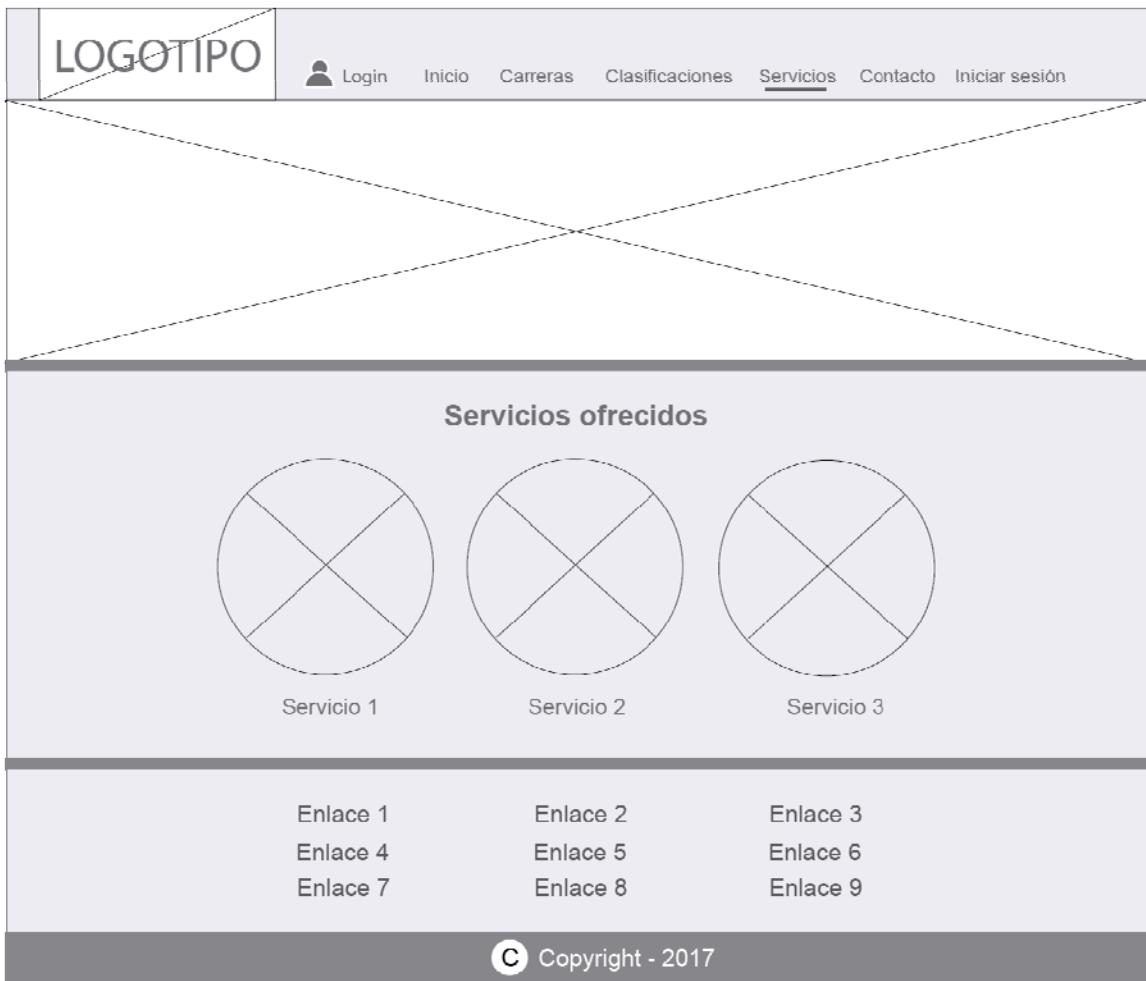
Enlace 1	Enlace 2	Enlace 3
Enlace 4	Enlace 5	Enlace 6
Enlace 7	Enlace 8	Enlace 9

© Copyright - 2017


## Página de clasificaciones



## Página de servicios



## Página de contacto

**LOGOTIPO**  [Login](#) [Inicio](#) [Carreras](#) [Clasificaciones](#) [Servicios](#) [Contacto](#) [Iniciar sesión](#)

---

### Formulario de contacto

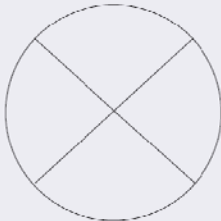
Teléfono

Mensaje

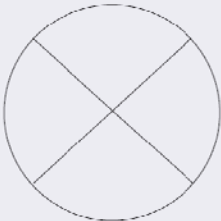
**Enviar formulario**

---

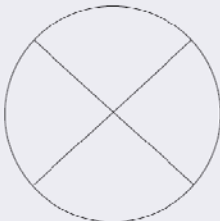
### Servicios ofrecidos



Servicio 1




Servicio 2




Servicio 3

---

Enlace 1	Enlace 2	Enlace 3
Enlace 4	Enlace 5	Enlace 6
Enlace 7	Enlace 8	Enlace 9

 Copyright - 2017

## Página de login

**LOGOTIPO**  [Login](#) [Inicio](#) [Carreras](#) [Clasificaciones](#) [Servicios](#) [Contacto](#) [Iniciar sesión](#)

---

### Login de usuario

Usuario

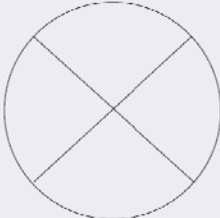
Password

**Iniciar sesión**

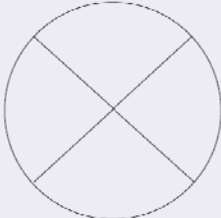
[Acceso a registro](#)

---

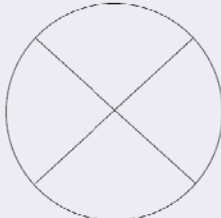
### Servicios ofrecidos



Servicio 1



Servicio 2




Servicio 3

---

Enlace 1	Enlace 2	Enlace 3
Enlace 4	Enlace 5	Enlace 6
Enlace 7	Enlace 8	Enlace 9

---

 Copyright - 2017