

GRADO EN INGENIERÍA INFORMÁTICA DE  
GESTIÓN Y SISTEMAS DE INFORMACIÓN  
**TRABAJO FIN DE GRADO**

***GESTIÓN DE INFORMACIÓN PARA  
PLATAFORMA EDUCATIVA BASADA  
EN TAREAS***

**Alumno/Alumna:** Fernández, Díez, Rubén

**Director/Directora (1):** Bilbao, Landache, Javier Jesús

**Director/Directora (2):** Bravo, Sevilla, Eugenio

**Curso:** 2018-2019

**Fecha:** 01/02/2019

# 0. Resumen

Durante este proyecto se han desarrollado y mejorado unas funcionalidades de un sistema existente que permite organizar concursos educativos cuyas preguntas los alumnos tienen que responder dentro de un tiempo establecido.

Los profesores que estén registrados en el sitio web tienen completa libertad de crear concursos y añadir alumnos de los centros que representen. Además, pueden visualizar las estadísticas de las respuestas a las preguntas que los alumnos han respondido. Por su parte, los alumnos pueden ver los resultados que han conseguido en los concursos que han realizado.

Las funcionalidades del sitio web se han desarrollado en PHP, aunque también se han utilizado ficheros JavaScript y HTML. Todo el sitio web tiene el soporte de una base de datos MySQL en la que se guarda toda la información necesaria para el correcto funcionamiento. También se puede destacar que este sitio web está operativo de manera que los usuarios podrán notar los cambios que se han realizado a lo largo del proyecto.

Entre las funcionalidades que se han desarrollado y mejorado, se pueden citar las siguientes:

- Registro.
- Estadísticas.
- Captcha.

# 0. Laburpena

Proiektu honetan dagoeneko existitzen den sistema bateko funtzionalitate batzuk garatu eta hobetu egin dira. Sistema horretan hezkuntza lehiaketak antola daitezke. Lehiaketetan parte hartzen duten ikasleek lehiaketa horietan egiten diren galderak erantzun behar dituzte, emandako denboran.

Webgune honetan izena ematen duten irakasleek lehiaketak antola ditzakete eta parte hartuko duten ikasleak gehitu ditzakete. Gainera, ikasleek erantzun dituzten galderetako erantzunak estatistiken bitartez aztertu ahal dituzte. Ikasleek, berriz, egin dituzten lehiaketen emaitzak ikusi ahal dituzte.

Webgunearen funtzionalitateak gehienez PHPn garatu dira, baina JavaScript eta HTML fitxategiak ere erabili dira. Webguneak erabiltzen duen informazioa MySQL datu-base bakar batean gordetzen da. Datu-base honi esker, funtzionalitate guztiak modu egokienean ibiltzen dira. Webgune hau jadanik martxan dagoela esanguratsua da. Hortaz, erabiltzaileek egin diren aldaketak nabarituko dituzte.

Garatu eta hobetu egin diren funtzionalitateen arteko batzuk honako hauek dira:

- Izena ematea.
- Estatistikak.
- Captcha.

# 0. Abstract

During this project, some functionalities have been developed and improved in an existing system where educational competitions are held. In these competitions, students must answer chosen questions in a given time.

In this website, registered teachers can create any type of questioning competition and they can add students from their schools. Moreover, once those students have answered questions, statistics are provided so that both teachers and students can see the results.

The aforementioned functionalities have been developed mostly in PHP, although some JavaScript and HTML files have also been used. The whole website is supported by a MySQL database where all the required information is stored for good performance. It is important to highlight that this website is operative, so users can notice any changes made throughout the project.

Some of the improved and developed functionalities are:

- Registration.
- Statistics.
- Captcha.

# Índice

<b>0. Resumen</b> .....	<b>2</b>
<b>1. Definiciones y vocabulario técnico</b> .....	<b>11</b>
<b>2. Introducción</b> .....	<b>13</b>
<b>2.1 Origen del proyecto</b> .....	15
<b>2.2 Descripción y situación del trabajo</b> .....	15
<b>2.3 Razones de elección del TFG</b> .....	16
<b>2.4 ¿Qué es <i>Bebras</i>?</b> .....	18
<b>3. Planteamiento inicial</b> .....	<b>19</b>
<b>3.1. Objetivos</b> .....	19
<b>3.2. Herramientas utilizadas</b> .....	20
<b>3.3. Alcance</b> .....	22
3.3.1 Aprendizaje.....	23
3.3.2 Organización .....	25
3.3.3 Captura de requisitos .....	27
3.3.4 Análisis y diseño.....	28
3.3.5 Implementación y desarrollo.....	29
3.3.6 Pruebas .....	30
3.3.7 Documentación .....	31
3.3.8 Resumen de la planificación realizada .....	33
<b>3.4. Planificación temporal</b> .....	<b>34</b>
<b>3.5. Gestión de riesgos</b> .....	<b>37</b>
3.5.1 Enfermedad o lesión.....	37
3.5.2 Problemas con el equipo con el que se trabaja .....	38
3.5.3 Problemas con las herramientas de desarrollo.....	39
<b>3.6. Evaluación económica</b> .....	<b>40</b>
3.6.1 Mano de obra .....	41
3.6.2 Gasto de software .....	41

---

3.6.3 Gasto de hardware .....	42
3.6.4 Gastos indirectos .....	43
3.6.5 Gastos totales .....	43
<b>3.7 Arquitectura .....</b>	<b>43</b>
<b>4. Análisis de antecedentes .....</b>	<b>45</b>
<b>5. Captura de requisitos.....</b>	<b>46</b>
<b>5.1 Jerarquía de actores .....</b>	<b>46</b>
<b>5.2 Casos de uso .....</b>	<b>47</b>
5.2.1 Usuario no identificado .....	48
5.2.2 Alumno .....	49
5.2.3 Profesor_Admin.....	50
5.2.4 Profesor .....	51
5.2.5 ProfesorResp_Admin.....	52
5.2.6 ProfesorResponsable.....	53
5.2.7 Administrador.....	53
<b>5.3 Modelo de dominio .....</b>	<b>56</b>
<b>6. Análisis y diseño .....</b>	<b>60</b>
<b>6.1 Estructura del sistema.....</b>	<b>60</b>
<b>6.2 Diagrama de secuencia .....</b>	<b>62</b>
6.2.1 Registro.....	62
6.2.2 Estadísticas .....	64
<b>7. Elección de herramientas y tecnologías.....</b>	<b>66</b>
<b>7.1 Elección de lenguajes de programación .....</b>	<b>66</b>
<b>7.2 Elección de herramientas.....</b>	<b>66</b>
<b>8. Desarrollo.....</b>	<b>69</b>
<b>8.1 Necesidades del cliente.....</b>	<b>69</b>
8.1.1 Registro.....	69
8.1.2 Estadísticas .....	70
8.1.3 Captcha .....	74

---

8.1.4 Exportar estadísticas .....	75
<b>8.2 Cambios en la base de datos.....</b>	<b>76</b>
<b>9. Verificación y evaluación .....</b>	<b>79</b>
<b>10. Conclusiones y trabajo futuro .....</b>	<b>82</b>
<b>10.1 Análisis entre planificación estimada y real.....</b>	<b>82</b>
10.1.1 Herramientas utilizadas.....	82
10.1.2 Alcance.....	83
10.1.3 Planificación temporal.....	86
10.1.4 Evaluación económica .....	88
<b>10.2 Líneas futuras .....</b>	<b>89</b>
<b>10.3 Licencias.....</b>	<b>90</b>
10.3.1 Programas.....	90
10.3.2 Preguntas.....	90
<b>10.4 Reflexión personal.....</b>	<b>90</b>

## Índice de figuras

<b>Figura 1. Diagrama EDT del proyecto.....</b>	<b>23</b>
<b>Figura 2. Diagrama EDT de aprendizaje.....</b>	<b>24</b>
<b>Figura 3. Diagrama EDT de organización.....</b>	<b>25</b>
<b>Figura 4. Diagrama EDT de captura de requisitos.....</b>	<b>27</b>
<b>Figura 5. Diagrama EDT de análisis y diseño.....</b>	<b>28</b>
<b>Figura 6. Diagrama EDT de implementación y desarrollo.....</b>	<b>29</b>
<b>Figura 7. Diagrama EDT de pruebas.....</b>	<b>30</b>
<b>Figura 8. Diagrama EDT de documentación.....</b>	<b>31</b>
<b>Figura 9. Diagrama Gantt inicial del proyecto.....</b>	<b>36</b>
<b>Figura 10. Esquema con la arquitectura del sitio web.....</b>	<b>44</b>
<b>Figura 11. Jerarquía de actores.....</b>	<b>46</b>
<b>Figura 12. Caso de uso de Usuario no identificado.....</b>	<b>48</b>
<b>Figura 13. Caso de uso de Alumno.....</b>	<b>49</b>
<b>Figura 14. Caso de uso de Profesor_Admin.....</b>	<b>50</b>
<b>Figura 15. Caso de uso de Profesor.....</b>	<b>51</b>
<b>Figura 16. Caso de uso de ProfesorResp_Admin.....</b>	<b>52</b>
<b>Figura 17. Casos de uso de Administrador.....</b>	<b>53</b>
<b>Figura 18. Modelo de dominio inicial.....</b>	<b>55</b>
<b>Figura 19. Diagrama de clases resultante.....</b>	<b>60</b>
<b>Figura 20. Diagrama de secuencia de registro.....</b>	<b>63</b>
<b>Figura 21. Diagrama de secuencia de estadísticas.....</b>	<b>64</b>
<b>Figura 22. Gráfico estadístico de alumnos que han realizado los concursos.....</b>	<b>71</b>
<b>Figura 23. Gráfico estadístico de número de respuestas correctas, incorrectas y no respondidas en los concursos.....</b>	<b>72</b>
<b>Figura 24. Tabla con el número de alumnos que han participado en los últimos 4 años.....</b>	<b>73</b>



---

<b>Figura 25. Tabla con el número de preguntas respondidas bien, mal y sin contestar en los últimos 4 años (1).....</b>	<b>73</b>
<b>Figura 26. Tabla con el número de preguntas respondidas bien, mal y sin contestar en los últimos 4 años (2).....</b>	<b>74</b>
<b>Figura 27. Tabla con el número de preguntas respondidas bien, mal y sin contestar en los últimos 4 años (3).....</b>	<b>74</b>
<b>Figura 28. Modelo de dominio final.....</b>	<b>77</b>
<b>Figura 29. Diagrama EDT final de implementación y desarrollo.....</b>	<b>83</b>
<b>Figura 30. Diagrama de Gantt final.....</b>	<b>87</b>

## Índice de tablas

<b>Tabla 1. Resumen de la planificación realizada.....</b>	<b>34</b>
<b>Tabla 2. Gastos totales del proyecto.....</b>	<b>43</b>
<b>Tabla 3. Planificación temporal final.....</b>	<b>86</b>
<b>Tabla 4. Gastos totales finales del proyecto.....</b>	<b>88</b>

# 1. Definiciones y vocabulario técnico

Antes de empezar con la introducción y el desarrollo del proyecto, es importante conocer algunos de los vocablos que se usarán a lo largo de este documento. A continuación, se va a mostrar dicho glosario de vocablos o palabras técnicas:

- **Apache:** Es un servidor HTTP de código abierto. Gracias a su integración con PHP y, a través de XAMPP, una herramienta que permite trabajar con páginas web a nivel local, se puede ver cómo se va a mostrar un sitio web antes de subirlo al servidor. Su dirección oficial es: <https://httpd.apache.org/>. [1]
- **Bebras:** Hace referencia al sitio web en el que se está trabajando en el proyecto. También soporta el sistema para la realización y gestión del concurso educativo. Su dirección es: <https://bebras.ehu.eus/>.
- **Bot:** Es un programa informático que realiza tareas de forma automatizada a través de Internet. En este caso, de todo lo que puede hacer un *bot*, interesa la parte en la que puede atacar el sitio web que se está desarrollando. Es por ello que se ha puesto un *Captcha*, para evitar que ralentice el sitio web al intentar crear muchos usuarios en poco tiempo. [2]
- **Captcha:** Son las siglas de *Completely Automated Public Turing test to tell Computers and Humans Apart*. Se trata de una prueba que los usuarios tienen que cumplimentar para poder verificar que una persona está haciendo esa operación. Los desarrolladores de las aplicaciones deciden en qué lugares quieren tener disponibles los *Captchas*, aunque lo más normal es tenerlos en sitios críticos como puede ser el registro de un usuario en un sitio web. [3]
- **Certificado SSL:** Es un documento electrónico que contiene una clave que está formada por un correo electrónico, la empresa a la que pertenece y su ubicación. Las siglas vienen del inglés (*Secure Sockets Layer*) y permite asegurar y encriptar información sensible. Si en la dirección URL de un sitio web se ve “HTTPS” significa que dicha página está utilizando este certificado. Para más información: <https://www.ssl.com/>. [4]

- **DataTable:** Es un *plugin* para la librería *JavaScript jQuery* que permite mejorar la forma de interactuar con las tablas, permitiendo ordenar, añadir o filtrar tablas de HTML con suma facilidad. [5]
- **FTP:** El significado de estas siglas es *File Transfer Protocol*. Consiste en un protocolo de transferencia de archivos entre sistemas que están conectados a una red TCP y que siguen la arquitectura cliente-servidor. En la sección 3.2 se verá que FileZilla utiliza este protocolo. [6]
- **HTML:** El significado de estas siglas es *HyperText Markup Language*. Es un lenguaje de programación que utiliza etiquetas para crear contenido. Por medio de un navegador, este lenguaje se ejecuta e interpreta para poder mostrar los elementos programados en un sitio web. En este proyecto se utilizará este lenguaje para visualizar e interactuar con el sitio web.
- **JavaScript:** Es un lenguaje interpretado que se utiliza en páginas web dinámicas. Normalmente se utiliza del lado del cliente, es decir, el cliente ejecuta sus operaciones desde un ordenador local. Este lenguaje se utiliza para hacer más interactivas las páginas web, ya que su integridad con HTML permite crear funcionalidades más avanzadas. En este proyecto se utilizará para hacer enlazar, cuando haga falta, HTML con JavaScript. Su dirección oficial es: <https://www.javascript.com/>. [7]
- **JSON o JavaScript Object Notation:** Es un formato de texto ligero utilizado para el intercambio de datos. Además, permite la integración con muchos lenguajes de programación como PHP, HTML o JavaScript, entre otros. Las preguntas interactivas de este proyecto se guardan en este formato.
- **MySQL:** Es un sistema de gestión de bases de datos que permite acceder a los datos que estén almacenados mediante código estructurado en forma de comandos. En este proyecto, y a través de phpMyAdmin, se utilizará una base de datos a la que hará falta acceder para obtener los datos que sean necesarios y mostrarlos en pantalla. Su dirección oficial es: <https://www.mysql.com/>.
- **Pop-up:** Hace referencia a la ventana que emerge de manera automática en la pantalla. Suele aparecer para ofrecer información al usuario como consecuencia de alguna tarea que haya hecho.

- **PHP** o *Hypertext Preprocessor*: Es un lenguaje de código abierto que sobre todo se utiliza en el desarrollo web y que puede ser incrustado en HTML. Además, se puede utilizar en los sistemas operativos más utilizados (Linux, Windows, macOS...), permite escribir aplicaciones de escritorio y crear *scripts* por parte del servidor o a través de líneas de comando, En este proyecto se utilizará para añadir funcionalidades al sitio web. Su dirección oficial es: <http://php.net/>. [8]
- **SFTP**: Es el protocolo que añade SSH a FTP.
- **SSH**: Las siglas vienen de *Secure Shell*. Es un paquete de software que permite un sistema de administración seguro y la transferencia de archivos entre redes inseguras. Para conseguir esto, se hace segura la conexión entre cliente y servidor, encriptando esta conexión. Para más información: <https://www.ssh.com/>.
- **TCP**: El significado de estas siglas es *Transmission Control Protocol*. Es un protocolo que garantiza que la información se pueda entregar entre ordenadores en orden de envío y sin errores. Estos ordenadores pueden usar TCP para crear conexiones a través de las cuales pueden enviar datos. [9]
- **Widget**: Programa informático, normalmente de tamaño pequeño, que permite el acceso a funciones utilizadas con frecuencia. Hoy en día estos pequeños programas se pueden ver en sitios web o en las barras laterales que hay disponibles en los blogs. [10]

## 2. Introducción

La tecnología ha cambiado las costumbres de las personas. Tal y como se dice al inicio del artículo [11], estos últimos años se ha visto como la introducción de nuevas tecnologías ha tenido un impacto considerable en la sociedad. Esto se aplica a las empresas, pero también se ha visto un cambio notable en la educación. Antes se trabajaba con pizarra, tiza y libros y ahora cantidad de centros han incorporado las tecnologías a sus aulas. Este cambio ha hecho que cambiara la forma en la que los alumnos estudian y aprenden nuevos conceptos. De esta manera, además de lo que se imparte en clase, los alumnos tienen la oportunidad de aprender y contrastar lo que se ha impartido en clase con lo que hay en Internet.

Internet ha abierto un sinfín de puertas al resto del mundo y, si bien es cierto que no es conveniente entrar por muchas de estas puertas, hay muchas que se pueden utilizar para nuestro beneficio. Por ejemplo, las puertas de la educación. Así, es posible unir a cantidad de alumnos, profesores y centros de distintos lugares permitiendo que realicen las mismas actividades. De estar bien organizado, solo tendrían que conectarse a través de Internet a una sitio web determinado, desde sus respectivos lugares, y una vez haber finalizado las actividades, podrían dar retroalimentación. Esta es muy importante, ya que permitirá que los alumnos, que en el futuro utilizarán esa misma página, tengan una mejor experiencia y mejorará el proceso enseñanza-aprendizaje. Además, si es posible podrán adquirir mayor conocimiento de forma más entretenida y divertida.

Visto lo mucho que aporta Internet en general y a la educación en particular, ¿puede hacer que el proceso de aprendizaje de las distintas materias que se dan en la escuela sea más ameno y eficaz? Es muy probable que los centros sigan teniendo el mismo modelo de enseñanza que han tenido toda la vida. Esto sucede por diversos factores, entre ellos, la falta de presupuesto, la oposición por parte de algunos profesores a dedicar más tiempo y esfuerzo en nuevas tecnologías para facilitar el aprendizaje de las asignaturas que imparten, el hecho de que es muy difícil cambiar algo que lleva siendo igual desde hace tanto tiempo y el rechazo a siquiera cuestionar el modelo de aprendizaje que hay hoy en día.

Sin embargo, es un hecho que algunos de esos centros han mostrado interés en incorporar nuevos métodos de aprendizaje a los que ya tienen. Para adaptarse a los tiempos que corren, estos centros tienen sitios web en los que publican todo tipo de eventos relacionados con esos mismos centros. Entre esos nuevos métodos de aprendizaje están los sitios web dedicados a la educación. Con todo eso en mente se empezó este proyecto que presenta muy buenas ideas sobre la mesa, utilizando un sistema de concursos que contienen preguntas que los alumnos deberán responder en un tiempo determinado. Pero, ¿es posible utilizar esta idea y mejorarla para ofrecer una mejor experiencia a todos los usuarios que utilicen este sitio web?

El objetivo del TFG es utilizar esta base como apoyo e intentar mejorar el sistema existente, de manera que la experiencia que el usuario tenga con el sistema sea más provechosa, más cómoda y más fácil. Asimismo, hay algunos aspectos que se pueden mejorar desde el punto de vista de la gestión, de manera que se reduzca el trabajo que tengan que hacer los administradores, o en caso de no poder reducir el trabajo, facilitarlo.

## 2.1 Origen del proyecto

Este TFG es la continuación de un proyecto que empezó con la publicación de la convocatoria de proyectos de investigación Saiotek, a finales de febrero de 2012. Saiotek es el nombre de la convocatoria en la que el proyecto que fue financiado por el Gobierno Vasco para la financiación de los Planes de Actuación que presentan los Agentes Científico-Tecnológicos para los ejercicios 2012 y 2013.

A raíz de esto, y con el paso de los años, los directores del TFG, Javier Bilbao y Eugenio Bravo, vieron la necesidad de actualizar o añadir algunas funcionalidades en el sistema desarrollado.

Por una parte, vieron que se podían añadir funcionalidades que facilitarían las experiencias de los usuarios al utilizar el sitio web. También vieron necesarias nuevas funcionalidades, para ofrecer a los usuarios la posibilidad de que pudieran tener los datos a mano, sin tener que identificarse en el sitio web.

Por otra parte, se le han exigido unas actualizaciones de seguridad al sitio web, con lo que era necesario hacer algunos cambios para que las funcionalidades no se vieran afectadas por estos cambios. Esto también supone tener que revisar las funcionalidades para poder hacerlas de manera más sencilla o para corregirlas, pues podrían tener algunos agujeros de seguridad.

## 2.2 Descripción y situación del trabajo

Desde el punto de vista del usuario, este proyecto consiste en un sitio web que los profesores y alumnos de los distintos centros que se hayan registrado en el sistema pueden utilizar para realizar concursos educativos. Estos concursos se componen de ejercicios o preguntas que los profesores pueden proponer para que los alumnos realicen.

Las preguntas que los profesores formulen pueden ser de dos tipos:

- **Interactivas:** En este tipo de preguntas se exige que los alumnos respondan a las preguntas haciendo algo más que seleccionar la respuesta correcta. Esto puede implicar que el alumno tenga que escribir la respuesta correcta, arrastre objetos y los ordene de una manera concreta, seleccione alguna respuesta en la imagen para responder a la pregunta, complete algún puzle, escriba la respuesta correcta...

- **Opción múltiple (*multiple choice*):** En este tipo de pregunta es necesario que el alumno seleccione alguna de las respuestas disponibles o que se proponen.

Una vez terminados los concursos tanto alumnos como profesores tienen la posibilidad de ver los resultados de estos. En estos resultados se puede ver la puntuación que los alumnos han conseguido en los concursos y qué preguntas han respondido bien o no.

Los profesores, por su parte, además de poder ver cómo los alumnos de los que ellos son responsables han hecho el concurso o lo que cada alumno de ese curso ha hecho, pueden ver las estadísticas de esos concursos. En ellas tienen a su disposición gráficas que pueden manipular de manera que puedan representar los datos que ellos quieran. Asimismo, los profesores son los encargados de dar de alta en el sistema a los alumnos que vayan a hacer los concursos. Sin embargo, solo los profesores coordinadores de cada centro (a los que llamaremos profesores responsables del centro) pueden dar de alta a otros profesores en el sitio web, para poder repartir la gestión del trabajo.

También cabe mencionar que el sitio web dispone de un foro, en el que se pueden plantear problemas o cualquier tema relacionado con el concurso *Bebras*.

## 2.3 Razones de elección del TFG

Las razones por las que se elegí este TFG son las siguientes:

### Salir de la zona de confort

A lo largo de la carrera se ha hecho mucho énfasis en los lenguajes de programación orientados a objetos. Esto resulta útil para tener una idea de cómo se deben organizar las distintas clases o incluso el código que contienen esas clases. Sin embargo, en otras asignaturas que se imparten en la carrera, no hay ocasiones para aprender otros lenguajes de programación de una manera tan profunda como se aprende, por ejemplo, Java.

Esto cerró un poco mi mente a la hora de elegir TFGs que no tuvieran ese lenguaje de programación. A pesar de ello, fui consciente de que era poco probable que pudiera hacer un TFG desarrollado en ese lenguaje, con lo que decidí mirar otras opciones sin darle tanta importancia al lenguaje de programación.

### Curiosidad por PHP, JS y HTML



No le di tanta importancia al lenguaje de programación, pero cuando vi que había un TFG que involucraba utilizar PHP, JS y HTML llamó mi atención. Esto se debió a que en la carrera hubo algunas asignaturas en las que se vio muy por encima estos lenguajes y, al no haber profundizado mucho en ellos en aquel entonces, pensé que este TFG sería una buena oportunidad para meterse de lleno en ese mundo.

### **El proyecto está operativo**

A la hora de estudiar, una de las cosas que más me frustra es el hecho de que todo el rato hago tareas. Dichas tareas me sirven a mí, ya que adquiero conocimientos de las mismas, y son entregables para los profesores de las distintas asignaturas. Esto implica que el profesor recibe mi ejercicio, lo corrige, pone nota y fin. Sirve para conseguir una nota parcial y para aprender o consolidar algo impartido en clase. No obstante, haber realizado prácticas en una empresa en el cuarto curso cambió mi punto de vista. Realmente me satisfizo pensar el hecho de que un cliente de una empresa podría utilizar, en alguna ocasión, la funcionalidad que yo implementé.

Por lo tanto, sabiendo que este proyecto estaba operativo en forma de un sitio web que muchos usuarios estaban utilizando, pensé que sería una muy buena oportunidad para perseguir esa sensación. Quería intentar mejorar lo que el sistema ofrecía, de manera que mejorara la experiencia de los usuarios y facilitara el entendimiento de algunas funcionalidades que, en mi opinión, no dejaban claro qué significaba lo que estaba en pantalla.

## 2.4 ¿Qué es *Bebras*?

Como se puede ver en [12], *Bebras* es una iniciativa internacional cuyo objetivo es promocionar las ciencias de la computación (ciencia computacional o informática) y el pensamiento computacional entre los alumnos de todas las edades que reciben educación en las escuelas. Los profesores que supervisan a los participantes pueden integrar el desafío de *Bebras* en su enseñanza. Este desafío se realiza en las escuelas a través de los distintos ordenadores o dispositivos móviles.

El pensamiento computacional, que se ha mencionado con anterioridad, implica utilizar un conjunto de técnicas y destrezas para resolver problemas que los ingenieros software usan para escribir programas y aplicaciones. El desafío *Bebras* promueve la destreza para resolver problemas, además de conceptos informáticos como la habilidad para desglosar una tarea compleja en componentes más simples, diseño de algoritmos, reconocimiento de patrones, generalización de patrones y abstracción.

Este desafío tiene lugar la segunda semana de noviembre a nivel mundial, aunque algunos países lo han extendido a dos semanas. Muchos países organizan actividades a lo largo de todo el año, por ejemplo, eventos en los que se premia a los participantes, segunda ronda del desafío, campus de verano, talleres de profesores, recogida de estadísticas e investigación.

Los administradores de los concursos oficiales son los encargados de escoger las preguntas que los alumnos tendrán que responder. Estas preguntas se escogen a partir de un total de preguntas que los distintos países que participan en esta iniciativa ponen en común.

## 3. Planteamiento inicial

A continuación, se van a describir los apartados correspondientes al planteamiento inicial del proyecto. Para ello, se describen los objetivos iniciales del proyecto, el alcance, la planificación temporal que en un principio se creía necesaria, las herramientas que se han utilizado, la gestión de riesgos, una evaluación económica inicial del proyecto y la arquitectura que representa al mismo, por parte del cliente y del servidor.

### 3.1. Objetivos

Los objetivos del proyecto se han dividido en dos secciones. Así, es posible comprobar que dichos objetivos se cumplen y se pueden discernir de manera más clara. Hay que destacar que el proyecto en el que se está trabajando está operativo, con lo que cualquier cambio que se haga puede ser sensible.

#### Sitio web

En esta sección se muestran los objetivos iniciales que se quiere que cumpla el sitio web.

- **Visibilidad:** Viendo cómo está estructurado el sitio web se tiene que ver que las funcionalidades y los resultados que estas producen se expresen de una manera fácil, visible y significativa. De esta manera, el usuario que esté navegando las distintas funcionalidades podrá recibir la información que está buscando.
- **Dar información al usuario:** A la hora de completar funcionalidades o intentar cosas que no se permitan, es importante que el usuario tenga información al respecto. Por lo tanto, si hace algo que no está permitido hay que darle información sobre eso, bien en forma de mensaje, bien usando *pop-ups*...
- **Funcionalidades que no fallen:** Habrá algunos casos en los que si no se cumplen ciertos requisitos se tendrán que mostrar errores o determinados mensajes. Sin embargo, al ser un sitio web operativo, hay que evitar mostrar mensajes no deseados o que las funcionalidades den errores.

#### Generales y de documentación

En este apartado aparecen los objetivos que se han planteado de cara a la documentación, tanto en la memoria como en el código.

- **Dejar el código documentado:** Es importante dejar las partes modificadas o incluidas bien comentadas. De esta manera, si alguien retomara el trabajo en el futuro, no tendría que dedicar mucho tiempo a entender el código, pues estará explicado de manera clara y concisa.
- **Proporcionar una base de todo lo que rodea al proyecto:** En el proyecto no solo está el sitio web, sino que también hay una base de datos de la que hay que coger los datos, así como muchos archivos .php o .js que forman parte de dicho sitio web. Siempre será útil que tengan nombres significativos y que la documentación de todo lo que rodea al sitio web esté bien explicado y estructurado en caso de que alguien hiciera cambios en un futuro.

### Funcionalidades

En este apartado aparecen los objetivos iniciales que se han propuesto de las funcionalidades de las que se habló al inicio del proyecto.

- **Registro:** Se quiere mejorar el registro de manera que se tengan que cumplir determinadas condiciones con tal de evitar repeticiones. Entre otras condiciones están tener que evitar que un colegio que esté inscrito en una provincia se pueda volver a registrar o que se tenga que pasar por un *captcha* para completar el registro.
- **Estadísticas:** Algunas estadísticas no muestran los datos deseados. Hacen falta modificaciones para que se muestren los resultados de las preguntas interactivas.
- **Otras:** Completar otras funcionalidades que no se hayan previsto en un inicio.

## 3.2. Herramientas utilizadas

Aquí se detallan las diferentes herramientas que se utilizarán para el desarrollo del proyecto. También se explica, de manera resumida, en qué partes del proyecto se va a utilizar cada una de las herramientas descritas.

- **Cisco AnyConnect Secure Mobility Client:** Es un programa que permite el acceso VPN (*Virtual Private Network*). En el proyecto se utilizará para realizar conexiones privadas al ordenador que está situado en la escuela y que contiene

las herramientas necesarias para conectar con el servidor. Su página oficial es: <https://www.cisco.com/>.

- **Conexión a Escritorio remoto:** Es una aplicación de escritorio que viene por defecto en Windows 10. Con ella, y sabiendo la IP del equipo remoto al que se quiere conectar, es posible conectarse y trabajar en un escritorio remoto desde cualquier ordenador como si fuera un ordenador propio. En este proyecto será necesario utilizar esta aplicación, porque el ordenador que tiene acceso al servidor con el que se trabajará es uno que está en la escuela.
- **Dia:** Es un programa gratuito que permite crear diagramas. En lo relativo a este proyecto, se utilizará para crear los diagramas que permitan explicar tanto el funcionamiento del sitio web, como el desarrollo del código. Su página oficial es: <http://dia-installer.de/index.html.es>.
- **FileZilla:** Es un administrador de sitios gratuito que permite crear una lista de sitios FTP o SFTP (estableciendo una conexión cifrada que utiliza el protocolo SSH). En este proyecto será el servidor en el que se guardarán todos los archivos necesarios para que el sitio web pueda funcionar debidamente. Su página oficial es: <https://filezilla-project.org/index.php>.
- **Google Chrome:** Es un navegador web desarrollado por Google que está disponible de manera gratuita y que ha derivado de diversos proyectos de código abierto. A la hora de investigar o de visualizar el funcionamiento del sitio web será la herramienta a utilizar por preferencia.
- **Microsoft Office Word 2007:** Es un programa que permite el procesamiento de textos. Se utilizará este programa para escribir la memoria del proyecto. La razón por la que se utilizará esa versión es porque es una versión que utilizo desde hace tiempo y porque tiene todo lo necesario para realizar la memoria.
- **Microsoft Office PowerPoint 2007:** Es un programa que permite realizar las presentaciones a través de diapositivas. Se va a utilizar para realizar la defensa del proyecto. La razón por la que se utilizará esa versión es porque es una versión con la que siempre he trabajado.
- **phpMyAdmin:** Es una herramienta gratuita programada en PHP que permite administrar MySQL en la web. En este proyecto se utilizará para guardar la base de datos que contiene todos los datos relativos al sitio web. Su página oficial es: <https://www.phpmyadmin.net/>.
- **Sublime Text 3:** Es un editor de texto preparado para la edición de código. Ofrece soporte para muchos lenguajes de programación, así como la habilidad

de resaltar la sintaxis para que se pueda saber dónde aparece en todo el código. Esta herramienta se utilizará para desarrollar el código del sitio web. Su página oficial es: <https://www.sublimetext.com/>.

- **XAMPP:** Es una distribución de Apache completamente gratuita que permite gestionar bases de datos MySQL e interpretar lenguajes como PHP. En este proyecto se utilizará para trabajar de manera local, para así poder subir los cambios y actualizaciones que haya que hacer una vez comprobado que funcionen. De esta manera, no hace falta subir constantemente los cambios al servidor y, desde el punto de vista del usuario que utiliza el sitio web, no interrumpirá nada de lo que esté haciendo en el sitio web. [1]

### 3.3. Alcance

Para explicar mejor este apartado se han realizado diagramas EDT para las distintas partes identificadas. Estas partes son: aprendizaje, organización, captura de requisitos, análisis de diseño, implementación y desarrollo, pruebas y documentación.

1. **Aprendizaje:** Este bloque contiene las tareas que están relacionadas con el estudio de las herramientas utilizadas en este proyecto. También hay que tener en cuenta el estudio de los lenguajes de programación empleados.
2. **Organización:** Este bloque hace referencia a las tareas que se han utilizado para que el proyecto se desarrolle correctamente.
3. **Captura de requisitos:** Este bloque contiene las tareas para identificar las funcionalidades a desarrollar en el proyecto.
4. **Análisis de diseño:** Este bloque representa las tareas que se han identificado para expresar de manera gráfica el código que se ha modificado o añadido en el proyecto.
5. **Implementación y desarrollo:** Este bloque explica las tareas que están relacionadas con el código que se haya utilizado para crear o modificar funcionalidades en el sitio web.
6. **Pruebas:** En este bloque se recogen las tareas que se han identificado con los respectivos métodos para comprobar el funcionamiento de las funcionalidades del sitio web.

7. **Documentación:** Este bloque corresponde a las tareas de redacción del proyecto. Además de la memoria en sí también hay que tener en cuenta otros lugares de documentación como pueden ser los archivos que contienen el código del sitio web.

Cada bloque de los mencionados anteriormente es predecesor al siguiente que está en la lista. Las excepciones a esto son el aprendizaje y la documentación, los cuales se van desarrollando a lo largo del proyecto y junto a los demás bloques.



Figura 1. Diagrama EDT del proyecto

La Figura 1 muestra la división de los bloques que se han planteado para este proyecto. Seguidamente, se explicará de manera detallada el contenido de cada uno de esos bloques.

Para todas estas tareas se ha hecho una pequeña descripción explicitando en qué consiste la tarea. Asimismo, se ha estimado de manera aproximada la duración de la misma, qué salidas o entregables se han hecho si es que los hay y los recursos que se han utilizado para llevar a cabo estas tareas.

### 3.3.1 Aprendizaje

Aquí se muestran las tareas que se han identificado relacionadas con el aprendizaje de las herramientas que se han propuesto en la Sección 3.2. Se van a omitir las herramientas que se conocían con anterioridad al proyecto, ya que a estas herramientas no se les ha dedicado ningún tiempo de aprendizaje a lo largo del proyecto.

La Figura 2 muestra el diagrama EDT resultante de las tareas del bloque de aprendizaje. Por una parte, están las herramientas nuevas que se han tenido que utilizar en el proyecto. Por la otra, están los lenguajes de programación que han sido necesarios para el avance del proyecto.

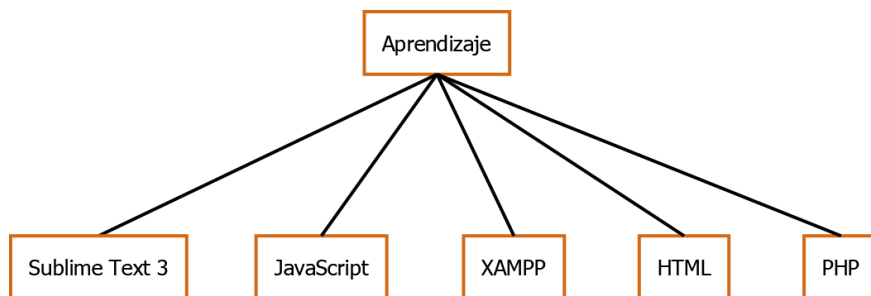


Figura 2. Diagrama EDT de aprendizaje

Las tareas relacionadas con las herramientas que se han estudiado son:

Sublime Text 3
<b>Bloque al que pertenece:</b> Aprendizaje.
<b>Duración estimada:</b> 5 horas.
<b>Descripción de la tarea:</b> Estudio del uso de la herramienta para programar y documentar el código de manera clara y concisa.
<b>Salidas/Entregables:</b> Ninguna.
<b>Recursos necesarios:</b> Sublime Text 3 y manual de Sublime <a href="https://www.sublimetext.com/">https://www.sublimetext.com/</a> .

XAMPP
<b>Bloque al que pertenece:</b> Aprendizaje.
<b>Duración estimada:</b> 5 horas.
<b>Descripción de la tarea:</b> Estudio del uso de la herramienta organizar el proyecto a nivel local, tanto el sitio web como la base de datos MySQL.
<b>Salidas/Entregables:</b> Ninguna.
<b>Recursos necesarios:</b> XAMPP, Versión PHP 5.2, Apache y MySQL.

Las tareas relacionadas con los lenguajes de programación que se han estudiado son:

JavaScript
<b>Bloque al que pertenece:</b> Aprendizaje.
<b>Duración estimada:</b> 15 horas.
<b>Descripción de la tarea:</b> Estudio del uso de JavaScript. Sintaxis que este lenguaje de programación utiliza e integración con PHP.
<b>Salidas/Entregables:</b> Ninguna.



<b>Recursos necesarios:</b> Sublime Text 3, Google Chrome y el sitio web <a href="https://www.w3schools.com">https://www.w3schools.com</a> .
--

PHP
<b>Bloque al que pertenece:</b> Aprendizaje.
<b>Duración estimada:</b> 25 horas.
<b>Descripción de la tarea:</b> Estudio del uso de PHP, además de la sintaxis que este lenguaje de programación utiliza.
<b>Salidas/Entregables:</b> Ninguna.
<b>Recursos necesarios:</b> Sublime Text 3, PHP 5.2 y el sitio web <a href="http://php.net/">http://php.net/</a> .

HTML
<b>Bloque al que pertenece:</b> Aprendizaje.
<b>Duración estimada:</b> 15 horas.
<b>Descripción de la tarea:</b> Estudio del lenguaje HTML. Ver la sintaxis que se usa y cómo integrarlo con PHP.
<b>Salidas/Entregables:</b> Ninguna.
<b>Recursos necesarios:</b> Sublime Text 3, Google Chrome y el sitio web <a href="https://www.w3schools.com">https://www.w3schools.com</a> .

### 3.3.2 Organización

Aquí se muestran las tareas que se han identificado relacionadas con la organización del proyecto.

La Figura 3 que aparece seguidamente detalla las tareas que se han identificado para este apartado.



Figura 3. Diagrama EDT de organización

A continuación, se van a detallar esas tareas:

Planificación de tareas a realizar
<b>Bloque al que pertenece:</b> Organización.
<b>Duración estimada:</b> 5 horas.
<b>Descripción de la tarea:</b> Identificación, explicación y desarrollo temporal de todas las tareas que tendrán que realizarse a lo largo del proyecto.
<b>Salidas/Entregables:</b> Diagramas EDT de los distintos apartados con las respectivas tareas a desarrollar.
<b>Recursos necesarios:</b> Día.

Selección de software
<b>Bloque al que pertenece:</b> Organización.
<b>Duración estimada:</b> 5 horas.
<b>Descripción de la tarea:</b> Identificar y valorar las distintas opciones de software que hay para luego usar la más útil o la que más ayude a lo largo del proyecto.
<b>Salidas/Entregables:</b> Documento con la lista de las aplicaciones que se van a utilizar.
<b>Recursos necesarios:</b> Microsoft Word 2007.

Instalación del software
<b>Bloque al que pertenece:</b> Organización.
<b>Duración estimada:</b> 7 horas.
<b>Descripción de la tarea:</b> Instalación y configuración de todas las herramientas que se van a utilizar a lo largo del proyecto.
<b>Salidas/Entregables:</b> Ninguna.
<b>Recursos necesarios:</b> Ordenador con conexión a Internet.

Reuniones con los directores del proyecto
<b>Bloque al que pertenece:</b> Organización.
<b>Duración estimada:</b> 10 horas.
<b>Descripción de la tarea:</b> La tarea hace referencia a las reuniones que se han hecho con los directores del proyecto a medida que este va avanzando. El objetivo de estas reuniones es tener un control periódico de

todo el trabajo realizado. Igualmente, se han debatido posibles mejoras que se pueden hacer o problemas que han podido surgir a lo largo del proyecto.
<b>Salidas/Entregables:</b> Documento con las explicaciones de las modificaciones o adiciones que se han hecho.
<b>Recursos necesarios:</b> Bolígrafo y cuaderno.

### 3.3.3 Captura de requisitos

Aquí se muestran las tareas que se han identificado relacionadas con la captura de requisitos hecha a lo largo del proyecto.

La Figura 4 muestra las distintas tareas que forman parte de la captura de requisitos aplicado a este proyecto.

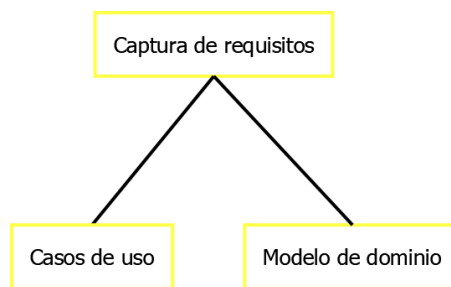


Figura 4. Diagrama EDT de captura de requisitos

Las tareas detalladas:

Casos de uso
<b>Bloque al que pertenece:</b> Captura de requisitos.
<b>Duración estimada:</b> 10 horas.
<b>Descripción de la tarea:</b> Identificar los distintos tipos de usuarios a los que va dirigida y que utilizan el sitio web.
<b>Salidas/Entregables:</b> Diagrama de casos de uso.
<b>Recursos necesarios:</b> Día.

Modelo de dominio
<b>Bloque al que pertenece:</b> Captura de requisitos.
<b>Duración estimada:</b> 5 horas.

<b>Descripción de la tarea:</b> Identificar las entidades que hacen falta para el modelo de dominio de la base de datos que se usa en este proyecto.
<b>Salidas/Entregables:</b> Diagrama de modelo de dominio.
<b>Recursos necesarios:</b> Día.

### 3.3.4 Análisis y diseño

En este apartado se muestran las tareas que se han identificado relacionadas con el análisis y diseño de las funcionalidades desarrolladas en el proyecto.

La Figura 5 muestra las distintas tareas que forman parte del análisis y diseño de las funcionalidades que pertenecen a este proyecto.

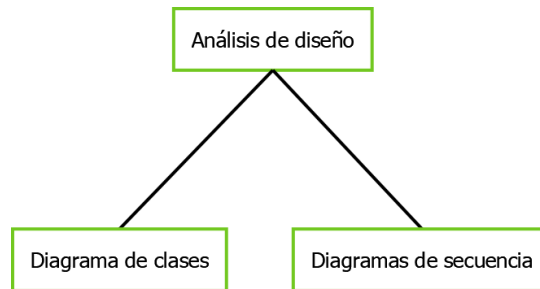


Figura 5. Diagrama EDT de análisis y diseño

Las tareas detalladas:

Diagrama de clases
<b>Bloque al que pertenece:</b> Análisis y diseño.
<b>Duración estimada:</b> 7 horas.
<b>Descripción de la tarea:</b> Identificar las clases que harán falta para cumplir con las funcionalidades exigidas. En este caso, al no haber clases como tal, el diagrama tendrá los métodos que se han utilizado para que las funcionalidades estén completas.
<b>Salidas/Entregables:</b> Diagrama de clases.
<b>Recursos necesarios:</b> Día.

Diagramas de secuencia
<b>Bloque al que pertenece:</b> Análisis y diseño.
<b>Duración estimada:</b> 15 horas.

<b>Descripción de la tarea:</b> Elaborar diagramas que describan las funcionalidades realizadas. Para ello, se describe minuciosamente qué hace el código en cada momento y de qué manera interactúa el usuario con el sitio web.
<b>Salidas/Entregables:</b> Diagramas de secuencia.
<b>Recursos necesarios:</b> Día.

### 3.3.5 Implementación y desarrollo

Se muestran las tareas que se han identificado que tienen relación con la implementación y el desarrollo del proyecto.

La Figura 6 muestra las tareas que se han identificado para este apartado, que coincide con las funcionalidades que se han exigido a lo largo del proyecto.

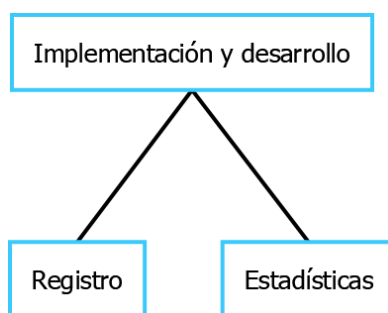


Figura 6. Diagrama EDT de implementación y desarrollo

Las tareas detalladas:

Registro
<b>Bloque al que pertenece:</b> Implementación y desarrollo.
<b>Duración estimada:</b> 15 horas.
<b>Descripción de la tarea:</b> Permitir que los usuarios se registren en el sistema. Además, se tienen que enviar mensajes tanto a la cuenta asociada de <i>Bebras</i> como al usuario que se ha registrado. El mensaje del usuario contiene el usuario y la contraseña para que se identifique y pueda usar el sitio web.
<b>Salidas/Entregables:</b> Fichero PHP con el código que permite ejecutar correctamente esta tarea.
<b>Recursos necesarios:</b> Sublime Text 3, Apache, FileZilla, MySQL, Conexión a Escritorio Remoto y ordenador con conexión a Internet.

Estadísticas
<b>Bloque al que pertenece:</b> Implementación y desarrollo.
<b>Duración estimada:</b> 15 horas.
<b>Descripción de la tarea:</b> Mostrar diversas estadísticas de interés a los distintos usuarios que utilizan el sistema. En algún caso el gráfico ya estaba, así que se ha comprobado si ese gráfico era de interés o expresaba bien los datos.
<b>Salidas/Entregables:</b> Fichero PHP con el código que permite ejecutar correctamente esta tarea.
<b>Recursos necesarios:</b> Sublime Text 3, Apache, FileZilla, MySQL, Conexión a Escritorio Remoto y ordenador con conexión a Internet.

### 3.3.6 Pruebas

Se muestran las tareas que se han identificado que tienen relación con las pruebas de las funcionalidades que se han desarrollado a lo largo del proyecto.

La Figura 7 muestra las tareas que se han identificado para este apartado.

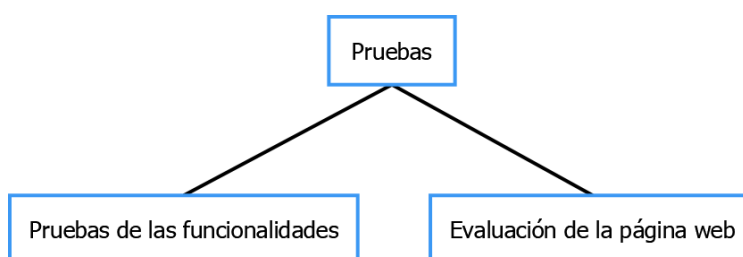


Figura 7. Diagrama EDT de pruebas

Las tareas detalladas:

Pruebas de las funcionalidades
<b>Bloque al que pertenece:</b> Pruebas.
<b>Duración estimada:</b> 30 horas.
<b>Descripción de la tarea:</b> Realizar las pruebas que sean necesarias para que las funcionalidades que se han pedido funcionen correctamente.
<b>Salidas/Entregables:</b> Ninguna.
<b>Recursos necesarios:</b> Sublime Text 3, Apache, MySQL y XAMPP.

Evaluación del sitio web
<b>Bloque al que pertenece:</b> Pruebas.
<b>Duración estimada:</b> 5 horas.
<b>Descripción de la tarea:</b> Todo lo que se ha desarrollado en el proyecto será utilizado por alumnos de muchas escuelas de España a través del sitio web bebras.ehu.eus. Aquí, se organizan concursos cuyas preguntas los alumnos tienen que responder para poner a prueba sus conocimientos. Es importante mencionar que el sitio web está operativo, con lo que la retroalimentación que se pueda obtener de los usuarios ayudará enormemente a mejorarlo.
<b>Salidas/Entregables:</b> Conclusiones que se saquen de los resultados.
<b>Recursos necesarios:</b> Retroalimentación de los usuarios a medida que las funcionalidades se completan.

### 3.3.7 Documentación

Se habla de las tareas que se han identificado que tienen relación con la documentación del proyecto. (Memoria y código del proyecto)

La Figura 8 muestra las tareas que se han identificado para este apartado.

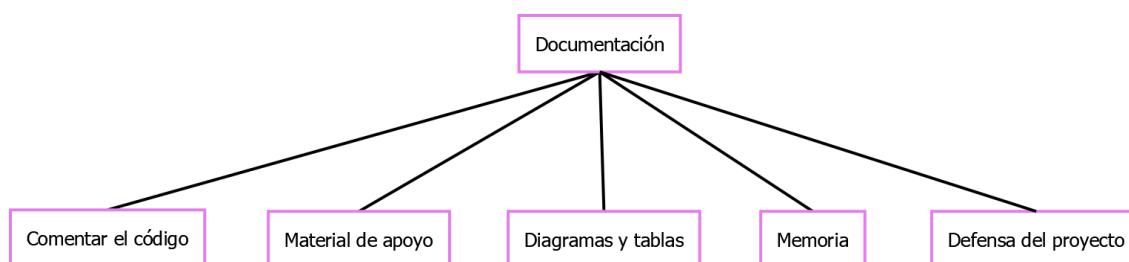


Figura 8. Diagrama EDT de documentación

Las tareas detalladas:

Comentar código
<b>Bloque al que pertenece:</b> Documentación.
<b>Duración estimada:</b> 15 horas.
<b>Descripción de la tarea:</b> Explicación de los trozos o líneas de código que se hayan modificado o añadido a lo largo del proyecto en los diversos

archivos.
<b>Salidas/Entregables:</b> Ficheros con el código documentado.
<b>Recursos necesarios:</b> Sublime Text 3.

Material de apoyo
<b>Bloque al que pertenece:</b> Documentación.
<b>Duración estimada:</b> 10 horas.
<b>Descripción de la tarea:</b> Búsqueda de material de apoyo (artículos, libros, documentos, memorias...) al que hacer referencia a la hora de explicar algunas partes que tengan que ver con el proyecto.
<b>Salidas/Entregables:</b> Bibliografía.
<b>Recursos necesarios:</b> Google Chrome y ordenador con conexión a Internet.

Diagramas y tablas
<b>Bloque al que pertenece:</b> Documentación.
<b>Duración estimada:</b> 8 horas.
<b>Descripción de la tarea:</b> Hace referencia a todas las tablas y diagramas que se hayan tenido que hacer a lo largo del proyecto, exceptuando los realizados en los apartados de captura de requisitos y análisis y diseño.
<b>Salidas/Entregables:</b> Tablas y diagramas.
<b>Recursos necesarios:</b> Dia, GanttProject y Microsoft Word 2007.

Memoria
<b>Bloque al que pertenece:</b> Documentación.
<b>Duración estimada:</b> 60 horas.
<b>Descripción de la tarea:</b> Redacción de la memoria del proyecto con todas las partes que lo integran.
<b>Salidas/Entregables:</b> Memoria del proyecto.
<b>Recursos necesarios:</b> Microsoft Word 2007.

Defensa del proyecto
<b>Bloque al que pertenece:</b> Documentación.
<b>Duración estimada:</b> 20 horas.
<b>Descripción de la tarea:</b> Realización de la defensa del proyecto.



<b>Salidas/Entregables:</b> Presentación del proyecto en forma de archivo PowerPoint.
---

<b>Recursos necesarios:</b> Proyecto, documentación y Microsoft PowerPoint.
---

### 3.3.8 Resumen de la planificación realizada

En la Tabla 1, que aparece a continuación, se muestra el resumen de las tareas definidas anteriormente. Junto a ellas, aparece el número de horas que se ha estimado para su realización, así como las tareas que han sido predecesoras, es decir, tareas cuya realización ha sido necesaria para poder completar la tarea indicada.

ID	Nombre de la tarea	Predecesores	Horas estimadas
	<b>Aprendizaje</b>		<b>65</b>
1	Sublime Text 3	-	5
2	XAMPP	-	5
3	JavaScript	-	15
4	PHP	-	25
5	HTML	-	15
	<b>Organización</b>		<b>27</b>
6	Planificación de tareas a realizar	-	5
7	Selección de software	6	5
8	Instalación de software	7	7
9	Reuniones con los directores del proyecto	-	10
	<b>Captura de requisitos</b>		<b>15</b>
10	Casos de uso	6	10
11	Modelo de dominio	6, 10	5
	<b>Análisis y diseño</b>		<b>22</b>
12	Diagrama de clases	11	7
13	Diagramas de secuencia	12	15
	<b>Implementación y desarrollo</b>		<b>30</b>
14	Registro	1, 2, 3, 4, 5, 10, 11, 12, 13	15
15	Estadísticas	1, 2, 3, 4, 5	15
	<b>Pruebas</b>		<b>35</b>
16	Pruebas de las funcionalidades	14, 15	30
17	Evaluación del sitio web	16	5
	<b>Documentación</b>		<b>113</b>

18	Comentar el código	14, 15	15
19	Material de apoyo	6	10
20	Diagramas y tablas	6	8
21	Memoria	7, 8, 18, 19, 20	60
22	Defensa del proyecto	21	20
	<b>TOTAL</b>	<b>21</b>	<b>307</b>

Tabla 1. Resumen de la planificación realizada

### 3.4. Planificación temporal

Ahora que ya se sabe qué tareas se van a realizar en el proyecto, se necesita establecer un espacio temporal en el que llevar a cabo dichas tareas. Para eso se ha realizado un diagrama Gantt, que permite distribuir las distintas tareas realizadas y ver el tiempo dedicado a cada una de ellas. Las tareas no son otras que las establecidas en el apartado de alcance del proyecto (Sección 3.3).

Para simplificar los cálculos temporales se va a suponer que se trabajan 4 horas diarias en el proyecto, a lo largo de 6 meses y medio en los que se va a trabajar en el proyecto. Cada mes tiene aproximadamente 4 semanas. Por lo tanto:

$$\text{Nº de meses} * \text{Nº de semanas} * \text{Nº de horas/semana} = 6,5 * 4 * 28 = 728 \text{ horas totales}$$

De estas 728 horas hay que quitar los días festivos, eventos, imprevistos y fines de semana, en los que no se va a trabajar salvo que sea necesario para que el proyecto siga avanzando dentro del tiempo establecido. Teniendo en cuenta todo esto, se ha estimado una pérdida de 100 horas. Así, las horas disponibles para poder trabajar en el proyecto son las siguientes:

$$\text{Nº de horas totales} - \text{Nº de horas perdidas estimadas} = 728 - 100 = 628 \text{ horas libres de trabajo.}$$

Considerando la estimación de 307 horas que se ha hecho (Sección 3.3.8), lo previsible es que se pueda entregar el proyecto a tiempo.

Por último, y tal y como se puede ver en el diagrama Gantt (Figura 9), hay algunas tareas que se han hecho en paralelo a otras, esto se ha hecho para poder tener toda la información en el momento en el que se realizan las distintas tareas. De

esta manera, se puede documentar todo lo que ha ocurrido de manera precisa, sin olvidar detalles que ocurrieron cuando se hizo la tarea correspondiente.

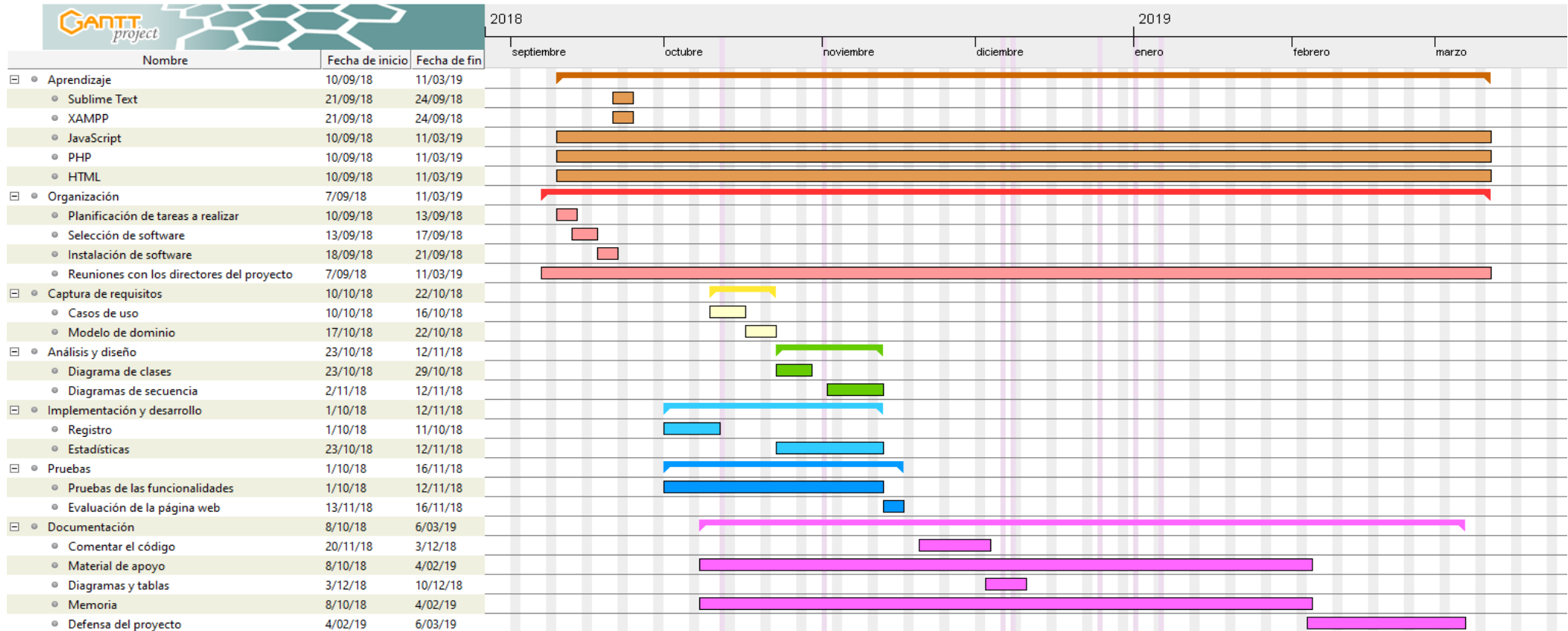


Figura 9. Diagrama Gantt inicial del proyecto

### 3.5. Gestión de riesgos

En este apartado se hablará sobre los riesgos que pueden ocurrir en la realización de este proyecto. Además, se hablará de un plan de prevención, de un plan de contingencia con las medidas a aplicar en caso de que ocurriera alguno de estos riesgos y de la probabilidad con la que pueden suceder dichos riesgos.

Hay que tener en cuenta que los diversos riesgos pueden tener distinta gravedad, por lo que impactarán de manera diferente al desarrollo del proyecto.

#### 3.5.1 Enfermedad o lesión

Trata sobre la indisponibilidad que puede ser causa del trabajo, salud u otro tipo de causas. Las consecuencias pueden ser tener que sufrir alguna enfermedad o que por consecuencia de algún accidente haya lesiones de diversa gravedad.

##### Prevención

- **Trabajo**
  - Al trabajar se debe mantener una correcta posición corporal, intentando adaptarse al entorno, iluminación, muebles, organización. Por supuesto, en todo momento hay que estar cómodo en el sitio de trabajo, manteniendo la distancia adecuada con el monitor.
  - Tomar descansos de manera periódica para que no afecte a largo plazo.
- **Salud**
  - Dormir las horas que se recomiendan dependiendo de la edad.
  - A ser posible, realizar algo de deporte siempre que no suponga riesgo.
  - Reposar en caso de tener fiebre o dolor de cabeza. Tomar alguna medicación si se confía en sus efectos.

##### Plan de contingencia

- Ir al médico en caso de tener alguna enfermedad o lesión para que se pueda diagnosticar y ver el alcance.

- Si este diagnóstico muestra que no había ningún síntoma se debe seguir con el trabajo. Sin embargo, y como medida de precaución, se descansará con más frecuencia o por periodos más prolongados que lo que se hacía con anterioridad.
- Si el diagnóstico muestra que realmente hay enfermedad o lesión se descansará el tiempo necesario para que el estado de salud no empeore. Por ello, no se volverá a coger el proyecto hasta estar bien de salud.

### **Probabilidad de que ocurra**

En mi experiencia personal, las probabilidades de que los casos anteriormente citados ocurran son las siguientes:

Probabilidad de enfermedad: 5%.

Probabilidad de lesión: 2%.

El tiempo necesario para la total recuperación dependerá de la gravedad de la enfermedad o lesión padecida. A mayor gravedad, mayor tiempo de recuperación.

## **3.5.2 Problemas con el equipo con el que se trabaja**

Este apartado hace referencia a los problemas que puedan surgir con el equipo de trabajo. Esto puede suceder porque haya ocurrido algún error con el equipo en sí, porque haya algún componente del equipo que esté dañado y que impida utilizarlo o porque el equipo de trabajo se haya infectado por algún virus.

### **Prevención**

- **Software**
  - Revisar el estado del equipo por medio de antiviruses o programas que permitan optimizar el estado del mismo.
  - Ejecutar funciones de limpieza de los distintos programas y del equipo para evitar que este se ralentice.
  - Efectuar copias de seguridad de forma habitual de los programas y del contenido realizado como mínimo. Almacenarlas en distintos lugares (discos duros externos, la nube...).

- **Hardware**

- Tener buena ventilación, para evitar sobrecalentamientos de los distintos componentes que formen el ordenador.
- Mantener el equipo limpio para que los ventiladores y otros componentes no cojan polvo y funcionen correctamente.
- Tener en cuenta el entorno en el que se trabaja para que el ordenador no se sobrecaliente.

**Plan de contingencia**

- **Software:** En caso de perder los datos o que alguna parte del sistema estuviera corrupta, se restaurará la última copia de seguridad realizada y se seguirá trabajando a partir de ella.
- **Hardware:** Si algún componente del ordenador fallara se intentará cambiar por otro que desempeñe su misma función y que esté en buen estado. De no ser posible el arreglo, se trasladará todo lo realizado a otro ordenador y se trabajará desde ese ordenador.

**Probabilidad de que ocurra**

En mi experiencia personal, las probabilidades de que los casos anteriormente citados ocurran son las siguientes:

Probabilidad de error software: 10%.

Probabilidad de error hardware: 5%.

El tiempo necesario para la total recuperación dependerá de las consecuencias que los errores de funcionamiento puedan provocar en el equipo. A mayor gravedad, mayor tiempo será necesario para restablecer el sistema a como estaba antes de que ocurriera el problema.

### 3.5.3 Problemas con las herramientas de desarrollo

A pesar de que la mayoría de las herramientas que se van a utilizar en el proyecto son conocidas, es necesario tener en cuenta posibles errores que puedan producir a lo largo del proyecto.

**Prevención**

- Conocer el funcionamiento de los programas que se vayan a utilizar en el proyecto.
- Utilizar páginas oficiales para descargar programas en los que se pueda confiar y para extraer todo tipo de información que pueda ser necesaria para el proyecto.

**Plan de contingencia**

- Utilizar Internet para solucionar los problemas que las herramientas pueden causar, pues es muy probable que otras personas hayan pasado por la misma situación. Leer manuales de los programas si fuera necesario.
- Usar los conocimientos que ya se tienen de esas herramientas y aplicarlos intentando buscar solución a los problemas.
- Sustituir las herramientas que se utilizaban si resultan no ser útiles porque hay herramientas que ofrecen mejores opciones.

**Probabilidad de que ocurra**

En mi experiencia personal, las probabilidades de que los casos anteriormente citados ocurran son las siguientes:

Probabilidad de problemas con herramientas: 40%.

El impacto temporal que supone la utilización de las herramientas es alto, porque ocurre con frecuencia que una herramienta no haga tanto como realmente se esperaba que hiciera. O viceversa, la herramienta hace lo que se espera que haga, pero el desconocimiento alarga el tiempo necesario para realizar la tarea. También ocurre que una herramienta deje de funcionar en un momento determinado por la razón que fuera, con lo que este tiempo que se pierde hay que tenerlo en cuenta.

**3.6. Evaluación económica**

Pese a que es un TFG del cual no se espera ningún beneficio económico, se va a hacer un planteamiento económico del proyecto.



### 3.6.1 Mano de obra

En la publicación oficial del BOE del 6 de marzo de 2018 [13] se puede ver que el sueldo mensual de un programador junior es de 1233,38 €. Para obtener el salario/hora, además del sueldo, se puede suponer que en una jornada laboral se van a trabajar 8 horas al día, 4 semanas al mes y 5 días a la semana y se puede calcular de la siguiente manera:

$$\text{Sueldo/hora} = \frac{\text{Sueldo mes}}{\frac{\text{semanas} * \text{días} * \text{horas}}{\text{mes} * \text{semana} * \text{día}}} = \frac{1233,38}{4 * 5 * 8} = \frac{1233,38}{160} = 7,70$$

Con lo que el sueldo/hora es de 7,70 € por cada hora trabajada.

A la hora de planificar la división temporal de las distintas tareas del proyecto, la duración estimada es de 307 horas. Sin embargo, de todas esas horas, 65 de ellas se van a emplear para aprender a manejar las herramientas y lenguajes de programación que se van a utilizar en el proyecto. Por lo tanto, se descuentan:

$$\text{Horas totales} - \text{Horas de aprendizaje} = 307 - 65 = 242 \text{ horas}$$

Por lo tanto, para saber el gasto de obra hay que multiplicar las horas que se trabajarán por lo que se pagará por hora:

$$\text{Horas trabajo} * \text{Sueldo/hora} = 242 * 7,70 = 1863,4$$

Con lo que el gasto económico por parte de la mano de obra es de 1863,4 €.

También se tiene que considerar el tiempo que los directores del proyecto han tenido que dedicar al proyecto. Se estimará que el sueldo/hora es de 10 €. Asimismo, las horas que se estima que empleen en este proyecto son 10. Por lo tanto,  $10 * 10 = 100€$ . Esto hace que el total se vea incrementado hasta los 1963,4 €.

### 3.6.2 Gasto de software

Por suerte, las herramientas que se van a utilizar en el proyecto son de uso gratuito con lo que no supondrá ningún tipo de gasto. No se tiene en cuenta el gasto producido por el servidor, porque la universidad es quien está proporcionando dicho servicio para desarrollar las funcionalidades del sitio web. Además, el sitio web

dispone de un certificado cuyo gasto y mantenimiento tampoco se va a tener en cuenta.

Las pruebas para el desarrollo de las funcionalidades se han hecho en un servidor aparte, lo cual supone un coste añadido.

Para este servidor de pruebas, y después de considerar diversos precios, se ha elegido un servidor de *hosting* básico del siguiente sitio web: <https://www.hostinet.com/hosting-web/>. Dicho servidor constaría 36,60 € por un año (a fecha de 28/01/2019), lo cual es tiempo más que suficiente para desarrollar las funcionalidades que se han desarrollado en este proyecto. Esto incluye el certificado SSL que permitiría hacer conexiones HTTPS al sitio web.

### 3.6.3 Gasto de hardware

En cuanto al gasto del hardware se refiere, se va a tener en cuenta el gasto de amortización de los aparatos electrónicos que se hayan utilizado a lo largo del proyecto, aunque en este caso solo se ha utilizado uno.

El único material electrónico utilizado ha sido un ordenador portátil HP Pavilion Notebook cuya vida es de aproximadamente 2 años y 5 meses. El coste fue de unos 1100 €. Para calcular la amortización mensual.

$$\text{Amortización mensual} = \frac{\text{Precio de coste}}{\text{meses de uso}} = \frac{1100}{29} = 37,93$$

Dando como resultado 37,93 €. Este ordenador se va a utilizar por 6 meses y medio que es la duración del proyecto. Por lo tanto:

$$\text{Amortización total} = \text{Amortización mensual} * \text{meses} = 37,93 * 6,5 = 246,5$$

La amortización del ordenador cuesta 246,5 €.

### 3.6.4 Gastos indirectos

Estos gastos hacen referencia a los gastos que no son del proyecto en sí, pero que tienen que ver con él. El gasto en cuestión es el realizado por la luz e Internet que se han utilizado.

Este gasto varía en función de lo mucho o poco que se utilicen estos servicios, así como del coste que suponen tanto la luz como la tarifa de Internet. Por estas razones, se va a suponer que el coste de los gastos indirectos es de 250 €.

### 3.6.5 Gastos totales

Este apartado sirve para resumir todos los costes de los que se han hablado en los apartados anteriores. Como se puede ver en la Tabla 2 los gastos totales ascienden a 2749,8 €.

Tipo de gasto	Coste en euros
Mano de obra	1963,4 €
Gastos software	36,60 €
Gastos hardware	246,5 €
Gastos indirectos	250 €
<b>TOTAL</b>	<b>2496,5 €</b>

Tabla 2. Gastos totales del proyecto

## 3.7 Arquitectura

Se utiliza la arquitectura cliente-servidor. Esta arquitectura implica que nuestro ordenador (cliente) establece conexiones con otros ordenadores remotos, solicitando algún servicio. Estos ordenadores remotos son los servidores.

Cuando se utiliza un recurso en internet, bien sea una transferencia de un fichero, una consulta a la base de datos o la identificación en una página web, se diferencian dos partes:

- **Ciente:** Es quien inicia la solicitud. Como ya se ha mencionado anteriormente, solicita un servicio al servidor. También se encarga de gestionar la conexión con el ordenador servidor, de recibir los datos que haya solicitado y de mostrar esos datos en pantalla de manera que los usuarios puedan ver esa información solicitada. Los usuarios tienen en la interfaz todos los comandos disponibles que reflejan las funcionalidades que el servidor ofrece. En este proyecto, los usuarios serán quienes tendrán que registrarse, identificarse o realizar concursos, entre otras acciones.
- **Servidor:** La funcionalidad del servidor es responder las solicitudes que los ordenadores cliente hacen. Estas funcionalidades se ven cumplidas con los ficheros que contienen el código que permite realizar esas solicitudes. En este proyecto, en el servidor se encuentran los ficheros con el código necesarios para que las funcionalidades que se ofrecen funcionen debidamente.

En la Figura 10 se puede ver todo lo mencionado anteriormente de manera resumida.

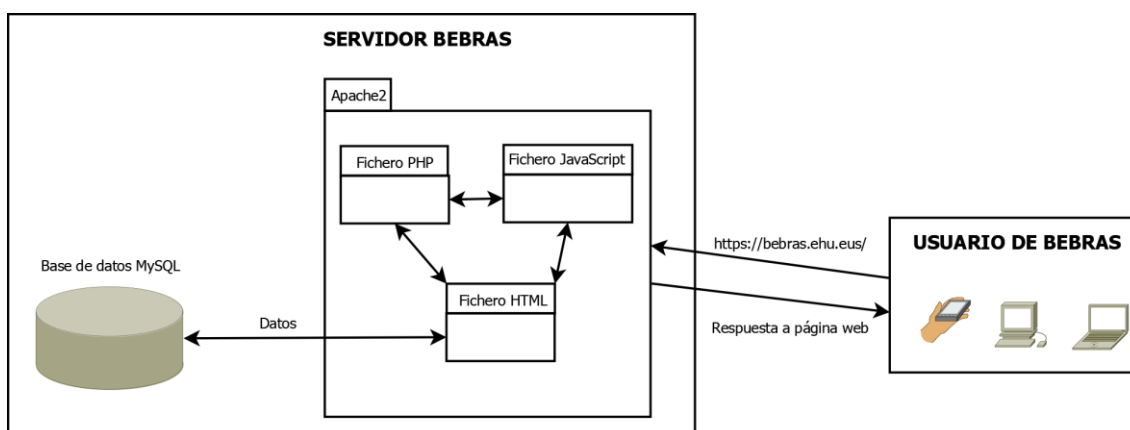


Figura 10. Esquema con la arquitectura del sitio web

Cada vez que el cliente acceda al sitio web, se hace una petición HTTPS al servidor de *Bebras*. Esta petición se recoge por Apache2 y el *script* PHP o HTML correspondiente se encarga de completar la tarea que el cliente haya solicitado. Sin embargo, aunque entre los ficheros PHP, HTML y JavaScript se puede intercambiar información, solo los de tipo PHP se pueden comunicar con la base de datos. Así, se puede introducir, extraer, modificar o borrar información dependiendo de la necesidad

## 4. Análisis de antecedentes

Como ya se ha mencionado anteriormente, este TFG es la continuación de un proyecto que empezó con la publicación de la convocatoria Saiotek, a finales de febrero de 2012. En marzo de ese mismo año, se concretó la propuesta para desarrollar una plataforma que los usuarios futuros de la misma utilizarían para mejorar las competencias en la enseñanza preuniversitaria. Dicha plataforma se creó para que los alumnos de primaria hasta bachiller de los centros que estuvieran de acuerdo con la propuesta tuvieran oportunidades de mejorar esas competencias.

Así, y con ayuda de los responsables de los centros que estaban dispuestos a ayudar en el desarrollo, se empezaron a proponer tareas que se alojaron en el sitio web [bebras.ehu.eus](http://bebras.ehu.eus) y que promovieron la competencia digital. Teniendo en cuenta el nivel académico para organizar la dificultad de las tareas, algunas de ellas son interactivas, es decir, permiten la acción, recogida de datos, toma de decisiones y reacción de los estudiantes que han participado mediante el sitio web. Las demás tareas consisten en preguntas de tipo test en las que los estudiantes tienen que seleccionar la respuesta que crean correcta.

Todo ese proyecto se hizo con el objetivo de mejorar la docencia en la enseñanza preuniversitaria en los centros del País Vasco, considerando la situación actual de los alumnos de primaria, secundaria y bachiller.

## 5. Captura de requisitos

En este capítulo se encuentran los requisitos que el sitio web desarrollado cumple durante el proyecto, para así satisfacer las necesidades de los usuarios.

En las siguientes secciones se mostrarán los diagramas relacionados con la captura de requisitos de las distintas funcionalidades que se hallan en el sitio web.

### 5.1 Jerarquía de actores

La jerarquía de actores consiste en establecer los privilegios que los diversos usuarios del sitio web gozan a la hora de utilizar la misma. A continuación, se podrá observar un esquema (Figura 11) con los distintos tipos de usuarios que forman parte del sistema así como una explicación detallada de lo que esos usuarios representan en ella.

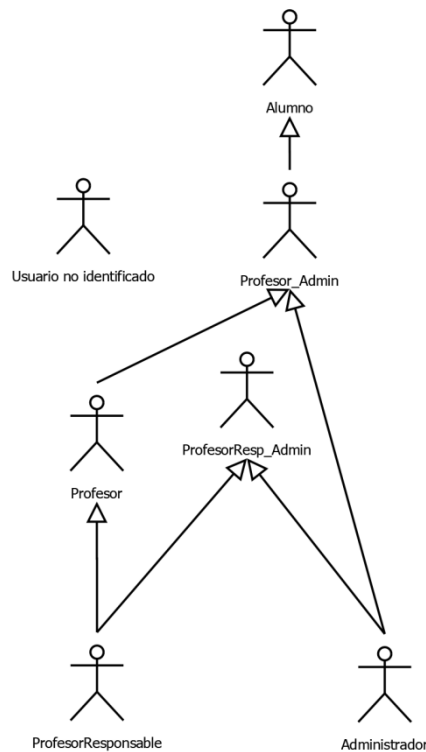


Figura 11. Jerarquía de actores

- **Usuario no identificado:** Este usuario corresponde al usuario de este sistema que no está identificado.
- **Alumno:** Representa a los usuarios identificados en el sitio web que sean alumnos de las distintas escuelas que formen parte de ella.
- **Profesor:** Representa a los usuarios identificados en el sitio web que sean profesores no responsables de las distintas escuelas que forman parte de la misma.
- **ProfesorResponsable:** Son los usuarios identificados en el sitio web que sean profesores responsables de las escuelas que forman parte de ella. Cada escuela solo puede tener un único profesor responsable.
- **Administrador:** Hace referencia a los administradores que gestionan el sitio web y organizan la información del mismo.
- **Profesor\_Admin:** Es un actor intermedio que no existe en el sistema como tal, pero que hace falta representar ya que “Administrador” y “Profesor” tienen funcionalidades en común, además de tener funcionalidades que el otro no tiene.
- **ProfesorResp\_Admin:** Es otro actor intermedio que no existe en el sistema como tal, pero que es necesario representar ya que “ProfesorResponsable” y “Administrador” tienen funcionalidades en común, además de tener funcionalidades que el otro no tiene.

## 5.2 Casos de uso

En este apartado se van a mostrar, con ayuda de diagramas, todas las posibles funcionalidades que los distintos tipos de usuarios descritos anteriormente pueden ejecutar en el sitio web. Además, se definirán las funcionalidades que pueden realizar.

### 5.2.1 Usuario no identificado

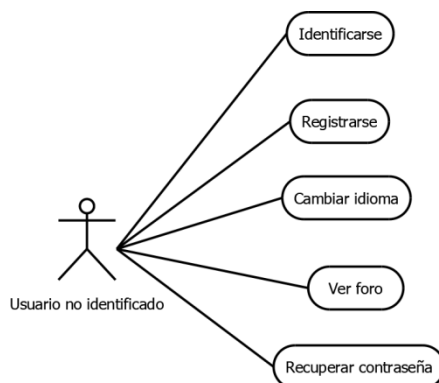


Figura 12. Caso de uso de Usuario no identificado

Las funcionalidades que “Usuario no identificado” puede hacer son las que se indican en la Figura 12. Ahora se explicarán dichas funcionalidades.

- **Identificarse:** Permite que este actor se identifique en el sitio web una vez proporcione el nombre de usuario y la contraseña. Para ello es necesario que esté registrado previamente, de manera que los datos para identificarse están ya en la base de datos.
- **Registrarse:** Permite que este actor se registre en el sitio web. Para ello, es necesario rellenar todos los campos que se ven en el formulario. Esos campos son los siguientes:
  - Campos relacionados con el centro: tipo, nombre, dirección, email, teléfono, provincia en la que se encuentra e idioma/s en los que se realizará la actividad en *Bebras*.
  - Campos relacionados con el profesor responsable: nombre, primer y segundo apellido, nombre de usuario, correo electrónico y cursos en los que ese profesor es responsable.
- **Cambiar idioma:** Permite que este actor cambie el idioma en el que se muestra el sitio web. Cabe destacar que una vez un usuario se identifique no podrá cambiar el idioma en el sitio web. Los idiomas disponibles son: castellano, euskera, catalán, gallego, inglés y francés.
- **Ver foro:** Permite acceder al foro que hay disponible. Los datos de registro en *Bebras* no tienen nada que ver con los del foro, con lo que aun estando registrado en *Bebras* no es posible la identificación en el foro usando los datos del sitio web de *Bebras*.



- **Recuperar contraseña:** Permite recuperar la contraseña en el caso de que no se recuerde. Se le enviará al usuario un correo electrónico con la contraseña al correo con el que tenga asociada la cuenta.

### 5.2.2 Alumno

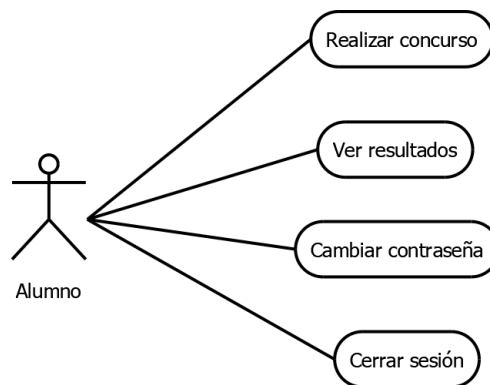


Figura 13. Caso de uso de Alumno

Las funcionalidades que “Alumno” puede hacer son las que se indican en la Figura 13. Ahora se explicarán dichas funcionalidades.

- **Realizar concurso:** Permite a los alumnos hacer concursos. Además, también se ofrece la posibilidad de realizar concursos a modo de entrenamiento. Para que un alumno pueda realizar un concurso, previamente un “ProfesorResponsable” tiene que activar ese concurso.
- **Ver resultados:** Permite ver los resultados que se han conseguido en los distintos concursos que se hayan realizado.
- **Cambiar contraseña:** Permite la contraseña que un usuario haya establecido para esa cuenta.
- **Cerrar sesión:** Permite cerrar la sesión y salir del sitio web.

### 5.2.3 Profesor\_Admin

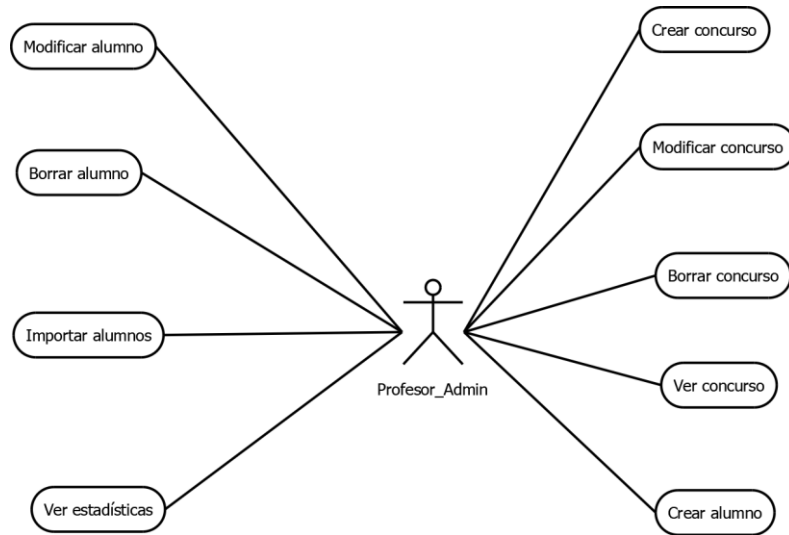


Figura 14. Caso de uso de Profesor\_Admin

Las funcionalidades que el “Profesor\_Admin” puede hacer son las que se indican en la Figura 14, además de las que “Alumno” puede hacer. Ahora se explicarán dichas funcionalidades:

- **Crear concurso:** Permite crear concursos. Al crear un concurso se da la elección de que ese concurso sea para entrenar o sea un concurso oficial. Los datos a rellenar son el idioma, el tiempo para realizar el concurso, año, categoría y los años de las preguntas a mostrar.
- **Modificar concurso:** Permite modificar las características de un concurso. Las únicas modificaciones permitidas son la categoría, el idioma y el año del concurso.
- **Borrar concurso:** Permite borrar todos los concursos.
- **Ver concurso:** Permite ver las preguntas que contiene un concurso. También se puede hacer un filtro de los concursos de los que se quieran ver las preguntas. Los filtros se pueden hacer en función de la categoría, idioma o el año de los concursos.
- **Crear alumno:** Permite crear alumnos que luego podrán utilizar el sitio web. Se exigen los siguientes datos: nombre, apellidos, curso, grupo, fecha de nacimiento, categoría, sexo y correo electrónico.

- **Modificar alumno:** Permite modificar los datos del alumno que se seleccione. Los datos que se pueden modificar son: nombre, correo electrónico, nombre de usuario, clave, curso, grupo, fecha de nacimiento y categoría.
- **Borrar alumno:** Permite borrar el alumno que se seleccione de la lista de alumnos que aparece en pantalla.
- **Importar alumnos:** Permite importar alumnos al sitio web. Esta importación se hará a través de un archivo *Excel* que se importará con el formato de *Excel 97* (extensión *.xls*) o de *Excel 2007* ó *2010* (extensión *.xlsx*). Este archivo contendrá tantas filas como alumnos se quieran introducir en la base de datos. El archivo contendrá cinco columnas de formato de celdas “*General*”, excepto la tercera columna que tendrá formato de celdas “*Fecha*”.
- **Ver estadísticas:** Permite ver las estadísticas de los alumnos que han participado en los concursos a lo largo del tiempo. Se puede filtrar por concursos, idioma, categoría y año.

#### 5.2.4 Profesor

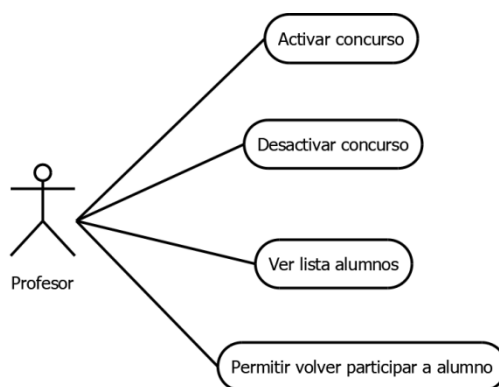


Figura 15. Caso de uso de Profesor

Las funcionalidades que “Profesor” puede hacer son las que se indican en la Figura 15, además de las que “Profesor\_Admin” puede hacer. Ahora se explicarán dichas funcionalidades:

- **Activar concurso:** Permite activar el concurso que se haya seleccionado. Además, aparecen los grupos de alumnos para los que está asignado ese concurso. De los grupos que se permita seleccionar es posible activar el concurso para que así los alumnos lo puedan realizar.

- **Desactivar concurso:** Permite desactivar el concurso que se haya seleccionado de los que hay disponibles.
- **Ver lista alumnos:** Permite ver el listado de alumnos de los diversos cursos a los que el profesor está asignado.
- **Permitir volver participar a alumno:** Permite que el alumno que se seleccione de los disponibles y que ya ha participado en un concurso vuelva a participar en el mismo.

### 5.2.5 ProfesorResp\_Admin

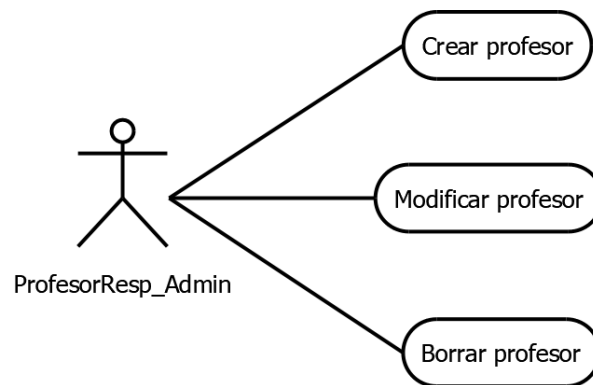


Figura 16. Caso de uso de ProfesorResp\_Admin

Las funcionalidades que “ProfesorResp\_Admin” hacer son las que se indican en la Figura 16. Ahora se explicarán dichas funcionalidades:

- **Crear profesor:** Permite crear profesores que no son responsables. Para ello es necesario proporcionar el nombre, el nombre de usuario, el correo electrónico, clave y el curso o cursos a los que pertenece.
- **Modificar profesor:** Permite modificar los datos de los profesores sean responsables o no. Los datos que se pueden modificar son: nombre de usuario, clave, correo electrónico y los cursos que tienen asignados.
- **Borrar profesor:** Permite borrar los profesores que se hayan seleccionado sean responsables o no.

## 5.2.6 ProfesorResponsable

El actor ProfesorResponsable no tiene casos que no sean los que tienen otros actores. Como se puede apreciar en la Sección 4.1, hereda los casos de uso de ProfesorResp\_Admin y Profesor.

## 5.2.7 Administrador



Figura 17. Casos de uso de Administrador

Las funcionalidades que el “Administrador” puede hacer son las que se indican en la Figura 17, también puede hacer todo lo que “Profesor\_Admin” puede hacer. Ahora se explicarán dichas funcionalidades:

- **Crear pregunta:** Permite crear preguntas que luego estarán en los concursos. Se pueden crear de dos tipos:
  - Interactiva: Este tipo de preguntas requiere que el usuario interactúe con el problema para poder resolverlo. Se exige identificador, categoría,

- año, título, nivel, idioma, marcar para saber si es pregunta obligatoria, marcar para saber si es pregunta visible, escribir unas palabras clave, escribir unos comentarios y adjuntar una imagen si es necesario.
- Tipo test: Este tipo de preguntas requiere que el usuario seleccione una de las múltiples respuestas que se ofrecen. Además de la información exigida en el punto anterior se exige una imagen y texto del ejercicio, indicar el número de respuestas que tendrá la pregunta, indicar la forma de mostrar las respuestas e indicar el texto e imágenes de cada una de las respuestas.
  - **Modificar pregunta:** Permite modificar la pregunta seleccionada. Para seleccionarla se puede filtrar por categoría, año o idioma. Los parámetros que se pueden modificar de la pregunta son identificador, categoría, año, título, nivel, idioma, si es pregunta obligatoria o no, si es pregunta visible o no, las palabras clave indicadas y los comentarios.
  - **Borrar pregunta:** Permite borrar la pregunta seleccionada. Primero hace falta buscar la pregunta que se quiera borrar, en función del idioma, categoría, nivel, si es pregunta obligatoria y el año. Una vez buscadas las preguntas que cumplen el criterio establecido se seleccionan aquellas preguntas que se quieran borrar.
  - **Duplicar pregunta:** Permite duplicar la pregunta seleccionada. Para seleccionarla se puede filtrar por categoría, año o idioma.
  - **Activar pregunta:** Permite activar las preguntas que se hayan creado para que sean elegibles para los concursos.
  - **Hacer visible concurso:** Permite hacer visible un concurso para que se pueda realizar.
  - **Ocultar concurso:** Permite ocultar un concurso.
  - **Cambiar puntuaciones:** Permite cambiar las puntuaciones que se otorgan para los distintos tipos de dificultades que tienen las preguntas. Las dificultades de las preguntas son: fácil, normal y difícil.
  - **Crear centros:** Permite crear centros a los que pertenecen los profesores y alumnos. Se exige el nombre del centro, la dirección, el teléfono, la provincia a la que pertenece el centro, el correo electrónico, el nombre del profesor responsable y el idioma o idiomas que se imparten en ese centro.
  - **Modificar centros:** Permite modificar los datos de un centro. Se pueden modificar los mismos datos que se necesitan para crear los centros.
  - **Borrar centros:** Permite borrar los centros que se hayan seleccionado de la lista.

- **Distribuir lista profesores:** Permite distribuir la lista de profesores responsables y no responsables que haya. Hace falta poner un título, contenido y si se quiere un adjunto antes de enviar.
- **Actualizar año alumnos:** Permite actualizar el año de los alumnos. Para ello solo hace falta seleccionar el alumno que se quiera actualizar.

### 5.3 Modelo de dominio

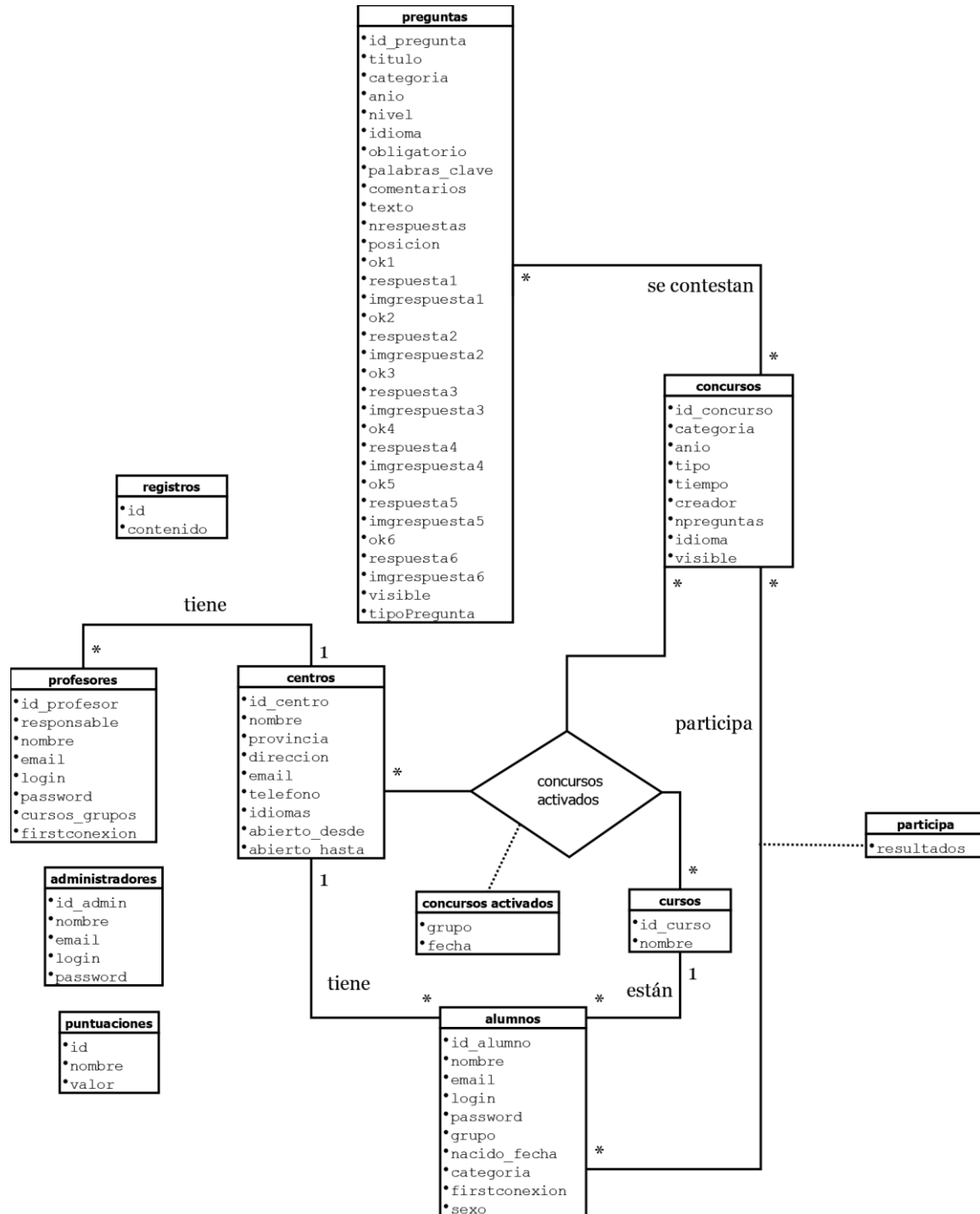


Figura 18. Modelo de dominio inicial



El modelo de dominio permite representar de manera visible las entidades, relaciones y atributos asociados a un problema planteado. También permite organizar la información que se guarda para que el sitio web pueda funcionar adecuadamente.

En la Figura 18 se muestra la primera versión con la que se ha trabajado en este proyecto. Al final se podrá ver que este modelo de dominio ha sufrido diversos cambios debido a las funcionalidades que se han exigido, de manera que algunas entidades que formaban parte del modelo de dominio inicial han perdido su utilidad.

A continuación, se explican las entidades que aparecen en la Figura 17:

- **Profesores:** Esta entidad representa a un profesor en el sitio web. Se guardan el id, si es responsable de *Bebras* del centro del que forma parte o no, el nombre, su correo (al que se le enviará un correo nada más registrarse al sitio web), login, password, los cursos de los que se va a encargar y si es la primera vez que se conecta o no al sitio web. En caso de ser la primera vez que se conecte a la misma, se le planteará si quiere cambiar la contraseña.
- **Centros:** Esta entidad representa un centro en el sitio web. Todos los datos que se guardan (salvo *abierto\_desde* y *abierto\_hasta* que son valores por defecto) vienen del formulario que el profesor responsable de *Bebras* de ese centro ha rellenado.
- **Administradores:** Esta entidad representa los administradores que están trabajando sobre el sitio web. Tienen todos los privilegios que el sitio web ofrece.
- **Puntuaciones:** Esta entidad representa las puntuaciones que las distintas preguntas que haya en el sitio web otorgan a los alumnos que las resuelvan. Así, además del id, se guarda el nombre de la dificultad (fácil, normal o difícil) para la que se otorga una puntuación concreta y dicho valor.
- **Alumnos:** Esta entidad representa los alumnos que forman parte del sitio web. Cabe destacar que sólo los profesores, tanto si son responsables de *Bebras* del centro como si no lo son, y los administradores pueden añadir alumnos al sitio web. Se exigen el nombre, email, login, password, grupo en el que están, fecha de nacimiento, la categoría en la que participarán los concursos (Benjamín, Cadete, Junior o Senior), la primera vez que se conecta al sitio web (como en la tabla “profesores”) y el sexo.
- **Cursos:** Esta entidad representa los cursos que se almacenan en el sitio web. Con guardar el id y el nombre del curso (1 primaria, 2 bachiller...) es suficiente.
- **Concursos:** Esta entidad representa los concursos que se realizan en el sitio web. Además del id, se guarda la categoría (mismos tipos que en la tabla

“alumnos”), el año en el que se celebra, el tipo (entrenamiento u oficial), tiempo que se tiene para finalizarlo, el creador, el número de preguntas que tiene, el idioma en el que se hará el concurso y si es visible o no para que lo realicen los alumnos.

- **Preguntas:** Esta entidad representa las preguntas que se almacenan en el sitio web. De las preguntas se almacena el id, título, categoría a la que pertenece, año en el que se ha creado, nivel de dificultad que tiene, idioma, si es una pregunta obligatoria de responder en el concurso, las palabras clave que tiene, los posibles comentarios que pueda tener, el texto que hace de enunciado de la pregunta, la posición en la que se ve la pregunta (vertical, horizontal o mosaico), si es visible o no en un concurso y el tipo de pregunta que es (test o interactiva). Además, al haber 6 respuestas como máximo se almacenan todas ellas con las imágenes correspondientes, los textos que corresponden a cada una de las respuestas y un indicador para saber si esa respuesta es correcta o no.
- **Concursos activados:** Esta tabla sale como resultado de la relación múltiple entre las tablas “centros”, “concursos” y “cursos”. Un profesor es el que puede activar un concurso. Por lo tanto, y viendo lo que hay en el modelo de dominio, al tener el profesor que activa el concurso, se puede saber qué centro ha activado, qué concurso y para qué curso. También se guarda para qué grupo se ha activado ese concurso y la fecha de activación. Para poder activar un concurso este tiene que estar visible previamente.
- **Participa:** Esta tabla sale como resultado de la relación binaria \*-\* entre las tablas “alumnos” y “concursos”. Aquí se almacenan los resultados que los alumnos han conseguido por cada concurso en el que hayan participado. Gracias a esta tabla, es posible sacar datos estadísticos que pueden ser de interés para los profesores y alumnos que estén en el sitio web.
- **Registros:** Esta entidad representa los registros que los usuarios han hecho en el sistema. Dispone de un id que diferencia los registros y también se guarda el contenido del registro en una cadena *string* que conforma el código HTML del registro.
- **Se contestan:** Esta tabla sale como resultado de la relación múltiple entre las tablas “preguntas” y “concursos”. No es necesario añadir ningún tipo de información ya que ambas tablas que forman parte de la relación tienen la información necesaria para el correcto funcionamiento de las funcionalidades que el sitio web ofrece.

Las relaciones que existen en este modelo de dominio son de 1-\* o \*-\*. En el caso de 1-\* significa que, por ejemplo, un centro tiene muchos profesores y que un profesor pertenece a un único centro. Para reflejar esto en la base de datos, se coge la clave primaria de la parte 1 y se coloca un atributo en la parte \* de la relación, lo cual indica que actúa como clave extranjera. En cuanto a \*-\* se refiere, que aparece en la relación “participa”, en un concurso pueden participar muchos alumnos y un alumno puede participar en muchos concursos. Por lo tanto, en la base de datos tiene que haber una nueva tabla que refleje ese hecho y que también tiene los atributos que se tengan que añadir a las claves primarias de las tablas que forman la relación.

Es importante aclarar que lo que se ve en la tabla “preguntas” no es práctico. Lo que habría que hacer sería crear una entidad llamada “respuestas” y relacionarla con “preguntas” a través de una relación 1-\*. De manera que, una pregunta tendría muchas respuestas y una respuesta pertenecería a una única pregunta. En “respuestas” se almacenarían todas las respuestas que una pregunta pudiera tener. En el caso de que una pregunta pudiera tener 50 respuestas, no es lógico crear tantos atributos por cada respuesta en la tabla “preguntas”, ya que sería impracticable tener que trabajar con una tabla que contuviera tantos atributos repetidos. En este caso en particular, al no tener muchas respuestas es posible tenerlo todo en una misma entidad. Así, no hace falta hacer búsquedas constantemente por el id de la pregunta a través de consultas MySQL que tengan en cuenta varias tablas. Es menos modular, pero al no tener muchos atributos por cada respuesta y, como cada pregunta no tiene muchas respuestas, no se hace pesado tenerlo todo en una misma tabla.

En el caso de los distintos tipos de usuarios que hay en el sitio web, hay que decir que se podría tener una única entidad con un atributo que indique el tipo de usuario que es. Sin embargo, los distintos tipos de usuarios tienen atributos que hace que sea imposible tenerlos localizados en una entidad única. Es por eso que están las entidades de “administradores”, “profesores” y “alumnos”.

## 6. Análisis y diseño

En este apartado se explicará el diagrama de clases que utiliza la parte en la que se ha trabajado, así como los diagramas de secuencia más importantes de las funcionalidades realizadas.

### 6.1 Estructura del sistema

En este sistema no hay un diagrama de clases como tal, ya que está enfocado a las funcionalidades que el sitio web puede realizar o no. Es decir, cada una de estas cajas representa los archivos y los métodos que estos utilizan. Ninguna de estas cajas representa entidades o conceptos como tal, aunque indican de qué funcionalidades se encargan. Esta estructura no es completa, solo hace referencia a la parte que se ha añadido o modificado.

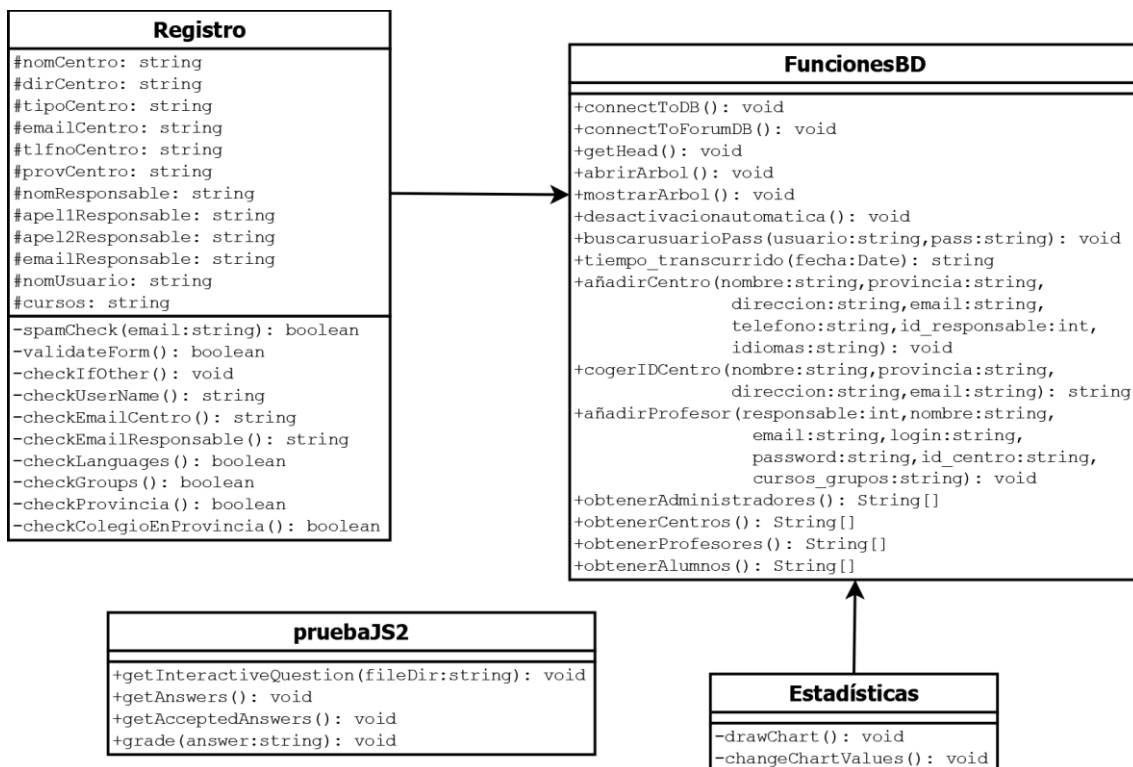


Figura 19. Estructura del sistema resultante

En la Figura 19 se puede ver cómo se han agrupado los métodos que se han necesitado para el funcionamiento correcto de las funcionalidades implementadas. Es importante mencionar que este diagrama no es completo, sino que es lo referente a lo que se ha realizado o modificado. Con lo que hay bastantes más funcionalidades que no están presentes.

Las distintas cajas hacen referencia a lo siguiente:

- **Registro:** Los atributos que utiliza esta caja son los que le vienen del formulario de registro. Los métodos, en cambio, se encargan de ejecutar distintas funciones que hay que exigir a los profesores que quieran registrarse en el sitio web.
- **FuncionesBD:** Esta caja contiene los métodos que han hecho falta para recoger información de la base de datos. Algunos de ellos también añaden información que luego se necesita almacenar en la base de datos. Es necesario mencionar que en algunos momentos puntuales se han hecho algunas llamadas a la base de datos, pero al ser tan triviales no se han añadido esos métodos. Debido a cómo se ejecutan algunas partes del código no ha sido posible modularizar estas operaciones.
- **pruebasJS2:** Esta caja se encarga de coger unos archivos .json predefinidos y los transforma en información útil. De aquí salen las preguntas interactivas de los distintos concursos que luego los alumnos tienen que responder. Una vez conocida la respuesta del alumno también se encarga de puntuar ese tipo concreto de preguntas.
- **Estadísticas:** Esta caja se encarga de dibujar las distintas gráficas que aparecen en las estadísticas que hay disponibles en el sitio web. Hay varios archivos que se encargan de hacer las estadísticas, cada uno con sus peculiaridades a la hora de mostrar los datos u ofrecer unos filtros determinados. Pero, todos ellos siguen el formato que se puede ver en el diagrama de clases.

Se ha mencionado que las clases no son clases como tal, sino que se representan las funcionalidades que se han implementado, junto a sus métodos. Sin embargo, hace falta mantener una cierta modularidad a la hora de programar, para así facilitar la introducción de nuevas funcionalidades al sitio web. Es por eso que se han separado las funcionalidades en esas cajas, para que cada una de ellas haga lo que tenga que hacer de esa funcionalidad sin estorbar a otras que ya estén.

## 6.2 Diagrama de secuencia

El objetivo de los diagramas de secuencia es representar de forma gráfica la manera en la que los objetos que hay en el sistema interactúan entre sí. De este proyecto, se han hecho los diagramas de secuencia de las funcionalidades que más importancia han tenido. En los diagramas de secuencia se puede apreciar que se utilizan clases que no se han definido en el diagrama de clases. Esto ocurre porque estas clases ya están definidas en JavaScript, con lo que simplemente se utiliza en el momento oportuno.

### 6.2.1 Registro

En la Figura 19 se puede ver el diagrama de secuencia final del registro. Es importante destacar que esta funcionalidad solo la pueden realizar aquellos usuarios que no se hayan identificado en el sistema.

Como se puede apreciar, el usuario tiene que rellenar los datos del formulario que aparece en pantalla. En algunos de estos campos se hacen ciertas comprobaciones como pueden ser:

- **checkUsername():** Comprueba que el nombre con el que ese usuario no identificado quiere identificarse a sí mismo en el sitio web no exista.
- **checkEmailResponsable() y checkEmailCentro():** Se comprueba que los emails que se proporcionan realmente sean emails.
- **checkIfOther():** Si en el desplegable de tipos de centro se selecciona “Otro”, aparece una nueva opción a rellenar.

Más tarde, cuando el usuario pulsa “Enviar” se hacen otras comprobaciones algo más complejas:

- **checkLanguages():** Comprueba que se haya seleccionado algún lenguaje de los que hay.
- **checkGroups():** Comprueba que se haya seleccionado algún grupo de los que hay.
- **checkProvincia():** Comprueba que se haya seleccionado una provincia de las que hay.
- **checkColegioEnProvincia():** Comprueba que en la provincia seleccionada ya haya un colegio registrado con ese nombre en la base de datos.

A partir de ahí, se envía un correo al profesor con un usuario y contraseña y otro a la cuenta de *Bebras*. Para finalizar el proceso, se añaden el centro y el profesor a la base de datos para que el profesor pueda identificarse en el sitio web.

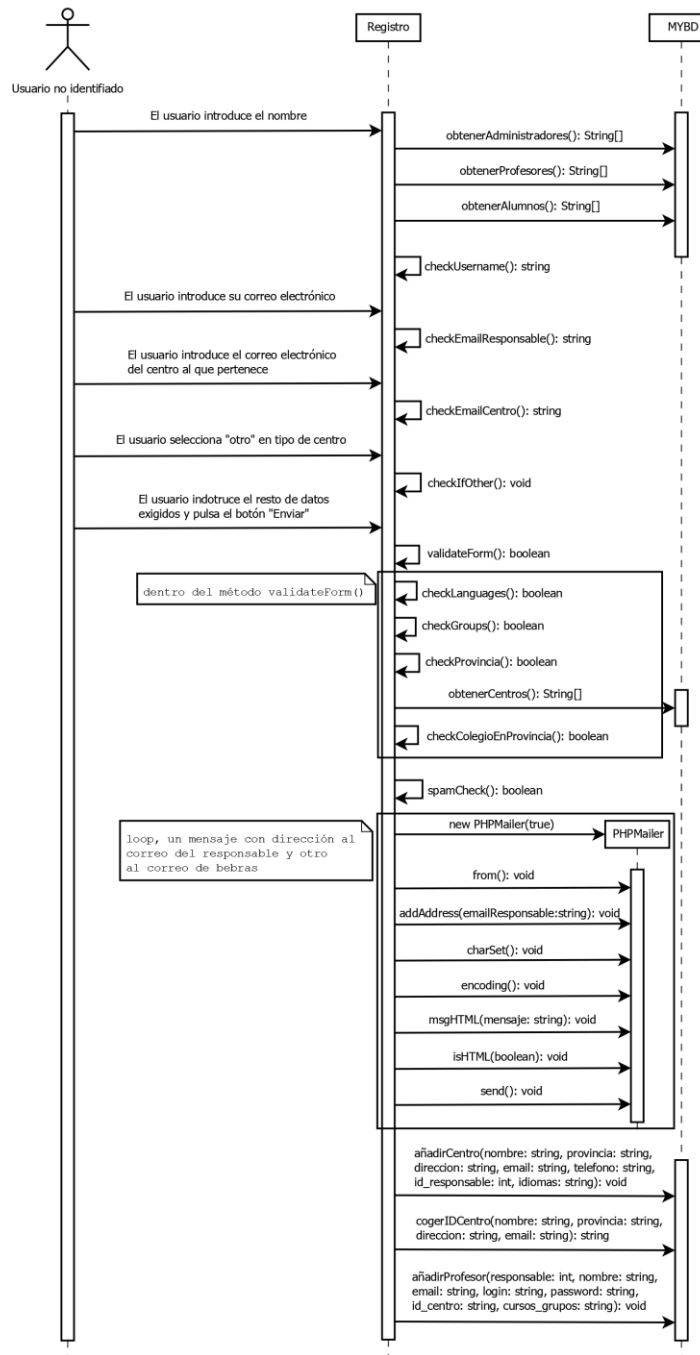


Figura 20. Diagrama de secuencia de registro

### 6.2.2 Estadísticas

En la Figura 20 se puede ver el diagrama de secuencia resultante de las estadísticas que se muestran en el sitio web. Cabe destacar que se muestran distintas estadísticas en varios puntos del sitio web, pero que todas ellas siguen el esquema que se observa en el diagrama de secuencia. Pero como la información que se maneja es diferente y la manera de enseñarlo también, ha sido necesario crear varios archivos de estadísticas, con lo que tienen sus variaciones. Este diagrama de secuencia es el que se utiliza para mostrar las estadísticas relacionadas con el número de alumnos que han realizado los concursos.

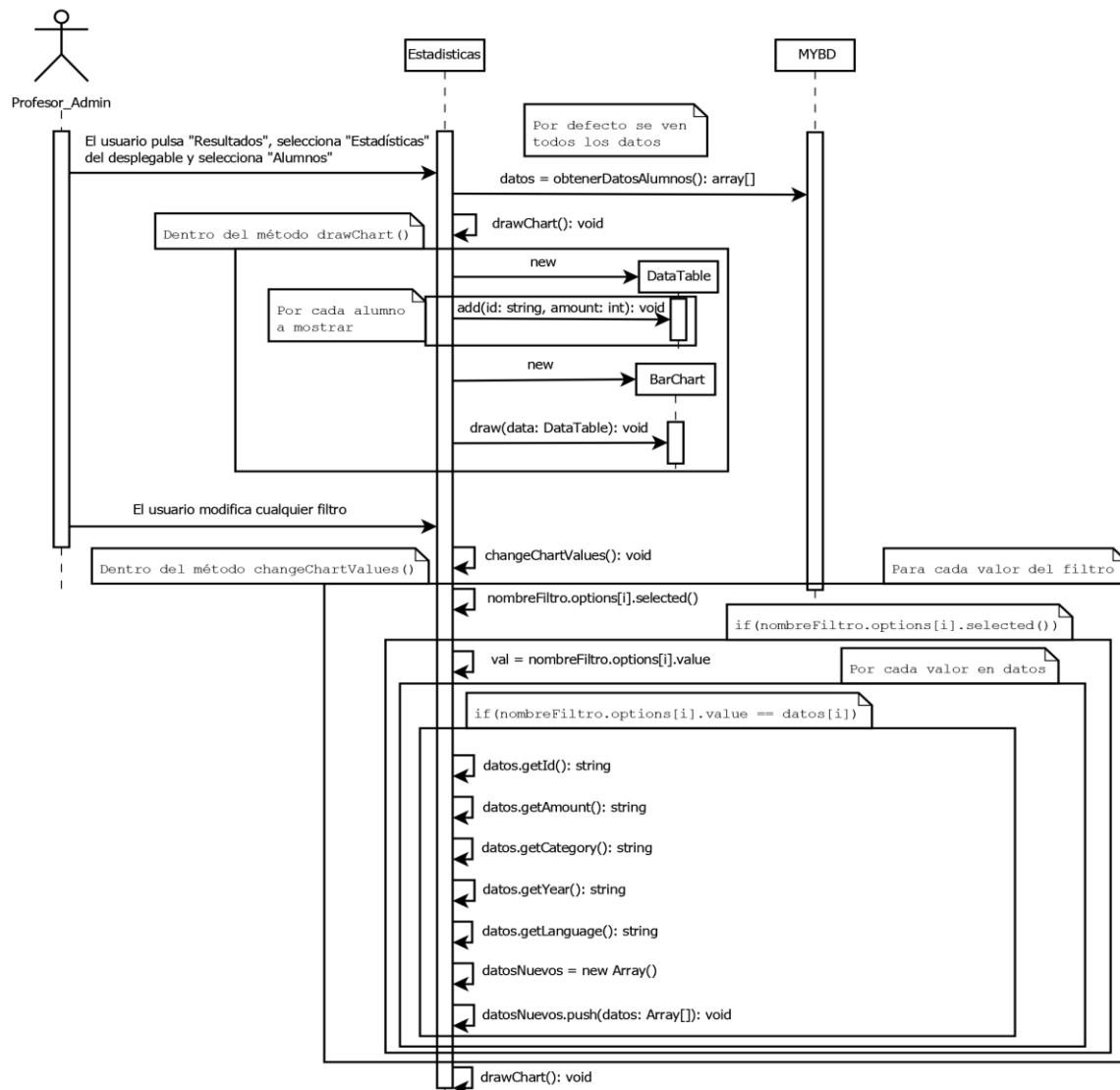


Figura 21. Diagrama de secuencia de estadísticas



En este diagrama se puede ver qué se necesita para mostrar las estadísticas de los alumnos, en este caso. Primero se obtienen de la base de datos los datos con los que se van a dibujar las estadísticas y se llama al método `drawChart()`. Este lo que hace es crear un *DataTable* en el que se añaden esos datos obtenidos, ya que los métodos que dibujan estas estadísticas lo hacen a partir de *DataTables*. Una vez que se tiene el *DataTable* con los datos preparados, se llama al método `draw()` de la clase predefinida *BarChart* para dibujar un diagrama de barras específicamente.

Con esto se dibuja el diagrama de barras pero sin alterar los filtros, con lo que se muestran demasiados datos y es imposible ver algo de utilidad. A medida que el usuario utilice los filtros ocurrirán cambios en los distintos gráficos que haya. Primero se comprueba cual es el valor del filtro que se ha seleccionado, después se comprueba si de todos los atributos disponibles alguno de ellos coincide en el *array* "datos" y, si coincide, se añade al *DataTable* para luego redibujar el gráfico.

Para poder crear las tablas de las estadísticas se cogen esos datos totales y se representan mediante tablas. Estas tablas salen de los resultados totales obtenidos, con los que también se hace la gráfica.

## 7. Elección de herramientas y tecnologías

Cuando se desarrolla un proyecto es muy importante escoger las herramientas necesarias y adecuadas. Esto permite que se puedan realizar más rápido las tareas en las que se tengan que usar esas herramientas, evitando así pérdidas de tiempo en los grupos que pueda haber en un proyecto. Esta pérdida de tiempo puede resultar en pérdidas económicas de diversa consideración dependiendo de lo importante que fuera dicha tarea.

En este caso, al tratarse de un proyecto en el que hay que codificar, es importante seleccionar correctamente los lenguajes de programación que hagan falta, así como las herramientas en las que se van a utilizar dichos lenguajes. Así, se podrá ahorrar mucho tiempo y trabajo innecesario.

### 7.1 Elección de lenguajes de programación

La elección de los lenguajes de programación fue bastante sencilla. El proyecto en sí exigía conocimientos en JavaScript, PHP, HTML y MySQL. Al trabajar con un sitio web que ya tenía una base realmente no tenía muchas más opciones que seguir con lo que este sitio web ofrecía. Principalmente se ha utilizado PHP, porque los archivos que más cambios han sufrido con las tareas exigidas a lo largo del proyecto han sido los .php. Bien es cierto que dentro de esos .php se utiliza JavaScript y HTML cuando se necesita, pero mayoritariamente se hace uso de PHP.

Por lo tanto, el lenguaje que se ha utilizado principalmente ha sido PHP, con ayuda de JavaScript y HTML cuando ha hecho falta.

### 7.2 Elección de herramientas

La elección de herramientas está basada en las herramientas seleccionadas en la Sección 3.2, con lo que se presentarán otras alternativas que se han considerado y las razones por las que finalmente se han descartado su uso en el proyecto.

- **Cisco AnyConnect Secure Mobility Client:** La razón principal por la que he escogido esta herramienta es que la propia universidad permite su descarga para conectarse a la red de la UPV/EHU. Para más información: <https://www.ehu.eus/es/web/ikt-tic/wifi-eduroam>.
- **Conexión a Escritorio remoto:** Para hacer conexiones remotas a otros dispositivos se han utilizado alternativas como TeamViewer. Sin embargo, tener que instalar una herramienta expresamente para realizar la conexión y tener que crear una cuenta lo hacen tedioso. Por ello, he terminado escogiendo la herramienta que Windows 10 ofrece para conectarse a escritorio remoto. Es sencilla, directa y sin complicaciones.
- **Dia:** A lo largo de la carrera, en las asignaturas de programación que hay en cada curso, a menudo se ha utilizado el Visual Paradigm. Al igual que Dia, este programa también resulta útil para realizar diagramas relacionados con la programación. Sin embargo, he preferido utilizar Dia porque es el que más he utilizado en la carrera y porque no sólo permite hacer diagramas relacionados con la programación, sino que también permite hacer diagramas de bases de datos, eléctricos, diagramas de flujo...
- **FileZilla:** Este programa es el que utiliza la UPV/EHU para conectar con el servidor y hacer modificaciones en los archivos. Por ello, no me he planteado utilizar otro que no fuera FileZilla.
- **Google Chrome:** Lo he utilizado por preferencia. Hay diversos navegadores que se pueden utilizar para esta tarea como pueden ser Mozilla Firefox, Opera, Internet Explorer, Microsoft Edge, Safari... No obstante, al haber trabajado mucho tiempo con este navegador y saber bien qué funcionalidades ofrece, he decidido seguir confiando en este navegador.
- **Microsoft Word 2007:** A pesar de que esta herramienta es un tanto anticuada, es la versión que tuve disponible cuando se realizó el proyecto. Otras alternativas podrían haber sido Overleaf, un editor de texto que permite escribir la memoria a través del lenguaje de LaTeX. Tener que aprender un nuevo lenguaje para tener que escribir un documento me echó hacia atrás y consideré otras alternativas. Entre ellas está Google Drive. En algunos trabajos grupales que he tenido que realizar durante la carrera, he tenido que adaptarme a las necesidades del grupo. Así, pude experimentar otra herramienta de edición de textos más, aunque lo poco cómoda e intuitiva que era no me gustó. Además, cuando el archivo de texto era muy largo (100+ páginas), vi que reaccionaba con muchísima lentitud, algo que enseguida hizo que descartara esta posibilidad.

Por lo demás, cualquier versión más reciente de Microsoft Word habría estado mucho mejor, de no ser porque es de pago.

- **phpMyAdmin:** Es la herramienta para acceder a la base de datos con la que trabajan en este proyecto así que decidí utilizarla. En cuanto al proyecto local se refiere, he utilizado la consola de comandos (Tecla Windows + R y después escribir cmd) para poder acceder a la base de datos local.
- **Sublime Text 3:** Tenía muchas dudas sobre esta herramienta, ya que yo siempre he utilizado Notepad++ y estaba muy satisfecho con lo que hacía. Pese a ello, compañeros de clase que utilizaron Sublime me lo recomendaron y decidí instalarla. Al final me gustó mucho y terminé utilizándola porque ofrece ciertas funcionalidades que Notepad++ no. Por ejemplo, en muchas líneas de código puede marcar de manera muy visible dónde empiezan y terminan paréntesis, llaves, corchetes... Además, en la parte superior derecha se puede utilizar el cursor y arrastrando de arriba abajo, hace el mismo efecto que el *scroll*, pero sin tener que arrastrar la barra espaciadora de la parte derecha, que suele ser difícil de utilizar por ser pequeña, o sin tener que utilizar la rueda del ratón. Resulta cómoda por su tamaño y porque permite moverse con facilidad en archivos muy grandes. También se pueden descargar *plugins* que permitan programar con mayor facilidad, y reconoce muchísimos lenguajes diferentes. Por si esto fuera poco, a la hora de escribir reconoce las variables que hay en el código, mientras que en Notepad++ solo reconoce las que se han escrito desde que se abrió ese archivo. Todos estos detalles, junto con otros que agilizan trabajar con muchas líneas de código, fueron decisivos para que me decantara por Sublime.
- **XAMPP:** Para poder trabajar a nivel local hay muchas aplicaciones que tienen utilidades similares, como por ejemplo, MAMP, WampServer o EasyPHP. En este caso, al haber utilizado XAMPP en otras asignaturas, decidí utilizar esta herramienta para trabajar en local.

## 8. Desarrollo

En este apartado se van a explicar las funcionalidades que se han ido desarrollando a lo largo del proyecto. Hay que destacar que han ido surgiendo nuevas necesidades por parte de los clientes Eugenio Bravo y Javier Bilbao, con lo que se ha tenido que hacer énfasis a aquellas funcionalidades que más urgieran.

### 8.1 Necesidades del cliente

Al hablar con los directores del proyecto se vio que el sitio web recibió una actualización de seguridad, con lo que hubo diversas funcionalidades que se vieron afectadas por esto.

#### 8.1.1 Registro

Lo que es el registro en sí estaba ya bien estructurado y operativo. El problema que surgió es que cuando se le daba a “Enviar” realmente no registraba al profesor ni el centro del que provenía a la base de datos. A raíz de esto se vio que había varios problemas con el registro.

Por una parte, el *captcha* que había en la página no funcionaba, lo cual impedía el registro. De esto se hablará más a fondo en la Sección 8.1.3.

Por otro lado, los campos del formulario tenían que cumplir ciertos requisitos que no cumplían. Lo lógico en un sistema que introduce colegios de distintas provincias para poder participar en los concursos es que si ya hay un colegio con un nombre determinado en una provincia concreta no se permita su registro. Sin embargo, esto antes no sucedía porque el registro se hacía a mano por parte de los administradores de la base de datos.

Para registrarse, tan pronto como el usuario pulsa “Enviar” se enviaba un correo al correo del sitio web con los datos del nuevo profesor que se quiere registrar. Una vez vistos los datos, se introducían manualmente en la base de datos tanto el centro como el profesor que se había registrado por parte de ese centro. Luego, se le enviaba a ese profesor un correo con los datos que había proporcionado y se le daba un usuario y contraseña para que pudiera acceder al sitio web.

El problema que tenía registrar a los profesores manualmente en la base de datos es que si se registraran muchos usuarios sería una labor bastante tediosa. Por esta razón, se vio necesario hacer el registro de manera automática, quitando mucho trabajo a los administradores.

En un principio funcionó bien, pero no mucho más tarde surgió otro problema. Los datos en la base de datos no seguían una codificación concreta con lo que al hacer ciertas comprobaciones de nombres de centros no se cogían los mismos valores. Es decir, si había algún centro en la base de datos con algún carácter especial (acentos, eñes, apóstrofes...) y se intentaba registrar en el sistema alguien de ese mismo centro, al no ser exactamente el mismo nombre, permitía el registro de ese centro en vez de impedirlo. Para solucionarlo, hubo que establecer una codificación específica a la hora de introducir los datos en la base de datos y a la hora de sacarlos de la misma, para así poder hacer la comparación y que diera el resultado esperado.

Los apóstrofes también fueron un pequeño problema. Esto ocurrió por cómo estaba el código del registro, y al no limpiar ese apóstrofe de los datos que el usuario introducía producía el error, porque el código pensaba que los apóstrofes estaban mal situados. Por ejemplo, al hacer la consulta:

```
“SELECT * FROM ordenador WHERE id = ‘id_ordenador’”;
```

si en la variable de entrada “id\_ordenador” hubiera algún apóstrofe en la cadena de caracteres que se recoge, causaría un error por el hecho de que el código no está preparado para ese apóstrofe, con lo que el apóstrofe inicial se cerraría con el introducido por el usuario.

### 8.1.2 Estadísticas

A la hora de representar las estadísticas, ha sido importante decidir qué estadísticas eran las más importantes a representar. Así, dependiendo de esas estadísticas, un gráfico u otro representarán mejor esa información.

Para saber qué gráfico escoger, hay que entender que hay variables discretas y continuas. Las variables discretas son variables que tienen un número finito de valores dados dos valores cualesquiera. Estos valores son números enteros y son el resultado de contar. Por ejemplo, el número de ovejas que hay en un rebaño o el número de coches que puede haber en una calle. Las variables continuas, en cambio, tienen un número infinito de valores dados dos valores cualesquiera. Estos valores son el

resultado de medir, por ejemplo, la longitud que una mesa puede tener o el peso de una persona.

Teniendo esto en cuenta y siguiendo los criterios mencionados en los artículos [14] y [15], se han tomado las siguientes decisiones para mostrar las siguientes estadísticas, entre otras:

- Número de alumnos que ha realizado un concurso:** Tal y como se expresa en el artículo de Sabadías anteriormente citado, si una variable es de carácter cualitativo, se pueden desarrollar diagramas de barras o rectángulos, en casos discretos. Dado que el número de alumnos que ha realizado un concurso es una variable discreta y hay distintos tipos de concursos, se ha decidido expresar los datos en una gráfica de barras horizontales. En la Figura 22 se puede ver que el eje de abscisas es el número de alumnos que ha realizado ese concurso, y que el de ordenadas contiene los tipos de concursos diferentes que han salido como resultado de haber seleccionado los filtros que hay disponibles.

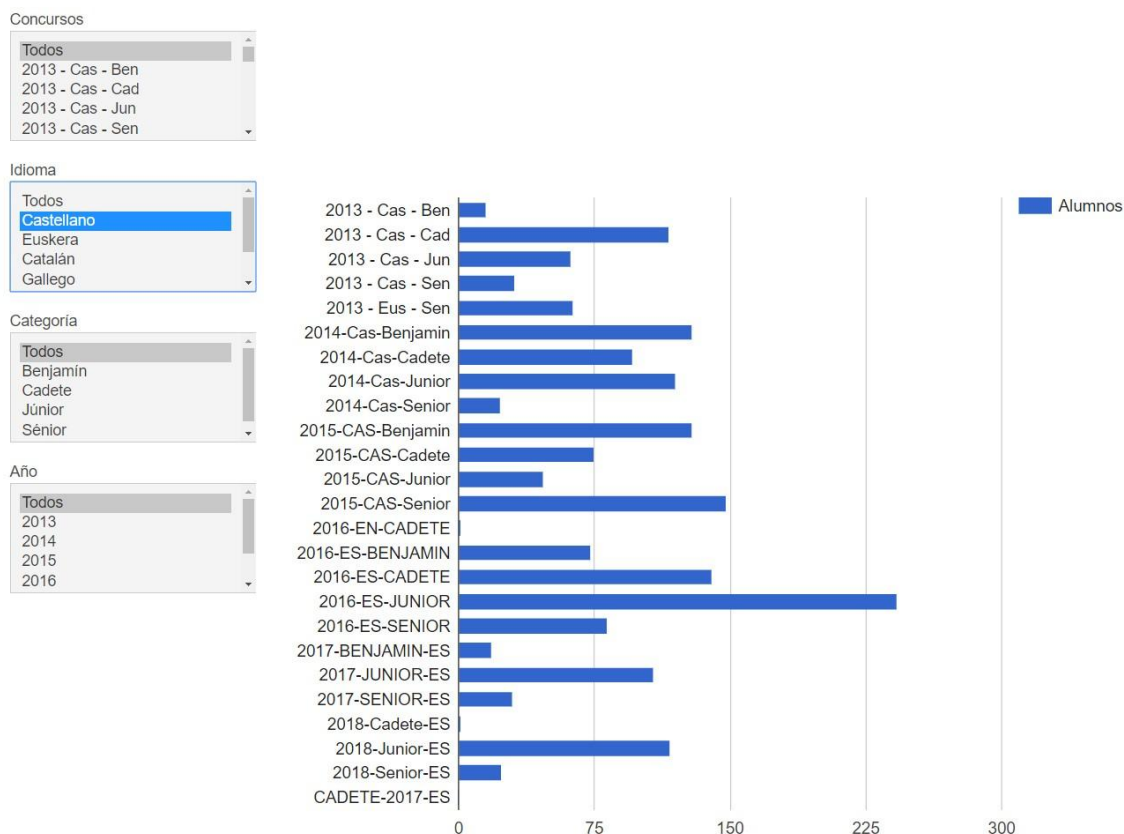


Figura 22. Gráfico estadístico de alumnos que han realizado los concursos

- Número de preguntas que los alumnos han respondido bien, mal o no han respondido:** En este caso, de todas las veces que se ha respondido esa pregunta, se han recogido las veces que se ha respondido bien, mal o no se ha respondido esa pregunta. Por lo tanto, se ha decidido hacer otro diagrama de barras, aunque verticales. Como se puede apreciar en la Figura 23, en el eje de abscisas se encuentran las preguntas que se hayan seleccionado como resultado de los filtros, y en el de ordenadas el número de veces que se ha respondido bien, mal o no se ha respondido la pregunta.

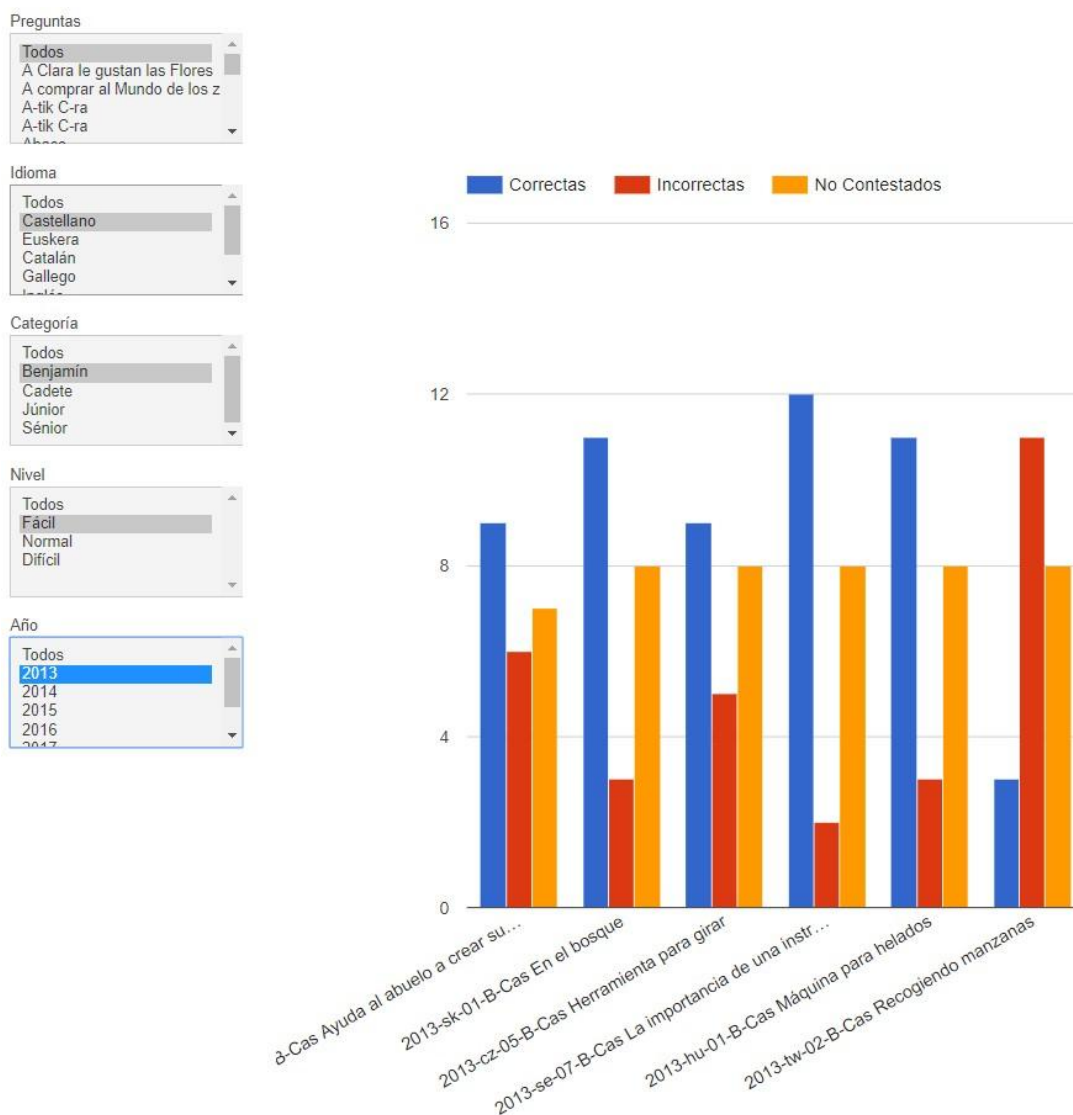


Figura 23. Gráfico estadístico de número de respuestas correctas, incorrectas y no respondidas en los concursos



Estas gráficas pueden expresarse de muchas maneras, aunque de la manera que se ha explicado, se pueden ver los resultados y hacer comparaciones de una manera visible.

A todo esto, se han añadido unas tablas debajo de los gráficos. Se han añadido en caso de querer ver los datos de una manera numérica. En las Figuras 24, 25, 26 y 27 se muestran tablas con las estadísticas de los resultados obtenidos en los últimos cuatro años en los que se han hecho concursos.

	Categoría 2018							Categoría 2017							Categoría 2016							Categoría 2015						
	T	EU	ES	EN	CA	GA	FR	T	EU	ES	EN	CA	GA	FR	T	EU	ES	EN	CA	GA	FR	T	EU	ES	EN	CA	GA	FR
Benjamín	0	0	0	0	0	0	0	18	0	18	0	0	0	0	73	0	73	0	0	0	0	129	0	129	0	0	0	0
Cadete	35	34	1	0	0	0	0	0	0	0	0	0	0	0	151	10	141	0	0	0	0	277	108	75	0	94	0	0
Júnior	210	93	117	0	0	0	0	114	5	108	0	1	0	0	395	110	242	25	18	0	0	121	52	47	22	0	0	0
Sénior	113	89	24	0	0	0	0	30	0	30	0	0	0	0	116	22	82	12	0	0	0	158	9	148	1	0	0	0
TOTAL	358	216	142	0	0	0	0	162	5	156	0	1	0	0	735	142	538	37	18	0	0	685	169	399	23	94	0	0

Figura 24. Tabla con el número de alumnos que han participado en los últimos 4 años

	Categoría 2018			Categoría 2017			Categoría 2016			Categoría 2015						
	OK	ER	NC	OK	ER	NC	OK	ER	NC	OK	ER	NC				
Benjamín	EUS	0	0	0	EUS	0	0	0	EUS	0	0	12	EUS	0	0	12
	ESP	0	0	0	ESP	68	182	14	ESP	352	503	81	ESP	655	932	153
	ENG	0	0	0	ENG	0	0	0	ENG	0	0	12	ENG	0	0	24
	CAT	0	0	0	CAT	0	0	0	CAT	0	0	0	CAT	0	0	0
	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0
	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0
Cadete	EUS	132	999	33	EUS	0	0	0	EUS	39	120	21	EUS	601	826	313
	ESP	0	3	9	ESP	11	4	0	ESP	526	1168	262	ESP	665	696	124
	ENG	0	0	0	ENG	0	0	0	ENG	8	4	12	ENG	0	3	12
	CAT	0	0	0	CAT	0	0	0	CAT	1	5	30	CAT	921	534	165
	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0
	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0

Figura 25. Tabla con el número de preguntas respondidas bien, mal y sin contestar en los últimos 4 años (1)

		OK	ER	NC		OK	ER	NC		OK	ER	NC		OK	ER	NC
Junior	EUS	125	978	61	EUS	73	22	31	EUS	658	652	202	EUS	795	1000	350
	ESP	185	1249	306	ESP	930	758	214	ESP	1192	1945	655	ESP	879	704	217
	ENG	0	0	0	ENG	0	0	0	ENG	135	161	4	ENG	208	277	175
	CAT	0	0	0	CAT	1	0	8	CAT	135	90	27	CAT	0	0	0
	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0
	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0
Senior	EUS	60	554	170	EUS	131	239	22	EUS	85	84	107	EUS	43	73	139
	ESP	82	496	30	ESP	772	552	132	ESP	236	452	332	ESP	914	971	515
	ENG	0	0	0	ENG	0	0	0	ENG	69	91	8	ENG	15	45	135
	CAT	0	0	0	CAT	0	0	0	CAT	0	0	0	CAT	0	0	0
	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0
	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0

Figura 26. Tabla con el número de preguntas respondidas bien, mal y sin contestar en los últimos 4 años (2)

		OK	ER	NC		OK	ER	NC		OK	ER	NC		OK	ER	NC
TOTAL PREGUNTAS POR IDIOMA	EUS	317	2531	264	EUS	204	261	53	EUS	782	856	342	EUS	1439	1899	814
	ESP	267	1748	345	ESP	1781	1496	360	ESP	2306	4068	1330	ESP	3113	3303	1009
	ENG	0	0	0	ENG	0	0	0	ENG	212	256	36	ENG	223	325	346
	CAT	0	0	0	CAT	1	0	8	CAT	136	95	57	CAT	921	534	165
	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0	GAL	0	0	0
	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0	FRA	0	0	0
TOTAL	OK	ER	NC	OK	ER	NC	OK	ER	NC	OK	ER	NC	OK	ER	NC	
	584	4279	609	1986	1757	421	3436	5275	1765	5696	6061	2334				

Figura 27. Tabla con el número de preguntas respondidas bien, mal y sin contestar en los últimos 4 años (3)

### 8.1.3 Captcha

Como ya se ha mencionado anteriormente el *captcha* es un elemento que permite evitar realizar una acción determinada de manera masiva. En este proyecto, se ha aplicado en el registro de los profesores y centros al sitio web. Por lo tanto, la funcionalidad es evitar que alguien cree usuarios manera masiva, pudiendo colapsar el procesamiento de los datos y ralentizar a otros usuarios que estén utilizando el sistema.

Al principio, el *captcha* que había era *reCAPTCHA*. Tal y como se explica en el artículo [16], cuando se visita un sitio web en el que hay un *reCAPTCHA*, se ve un

*widget* que recoge información acerca del navegador que utiliza el usuario y esta es enviada al servidor. Si el usuario responde correctamente el *reCAPTCHA*, se le permite seguir adelante mientras que si lo responde de manera equivocada tendrá que reintentar el *reCAPTCHA*.

Esta herramienta es válida, aunque de cara al futuro es fácil que pueda quedar obsoleta. Cuando queda obsoleta, deja de funcionar por muy bien que esté el código que la utiliza, lo cual era lo que sucedía e impedía el registro. Teniendo eso en cuenta, se ha escogido crear un propio *captcha* que cambie cada vez que se acceda o refresque la página en la que está localizado, en este caso, el registro. Por lo tanto, para que el usuario pueda registrarse es imprescindible que responda bien el *captcha* que se propone, que no es más que resolver una operación matemática sencilla. Así, se tiene un *captcha* que será eficiente a lo largo del tiempo, con lo que no hará falta preocuparse de las posibles versiones que puedan salir en el futuro y que puedan alterar el funcionamiento del *captcha* actual.

#### 8.1.4 Exportar estadísticas

Se ha comentado que en las estadísticas se han incluido unas tablas para poder ver los datos de una manera más numérica (Sección 8.1.2). Para poder tener esas tablas para posteriores análisis, se ha decidido dar la opción de exportar todas las tablas que aparezcan en las distintas estadísticas que contengan tablas. Las tablas se exportarán una vez se pulse el botón “Exportar” y se descargará un archivo *Excel* con los datos que se ven en las tablas. Es compatible con los buscadores Google Chrome, Mozilla Firefox, Microsoft Edge e Internet Explorer.

A la hora de exportar los datos ha habido varios obstáculos hasta llegar a la implementación que hay ahora. Uno de esos problemas consistía en que no se conseguía pasar la tabla con la codificación que estaba en la página. Esto se debió a que a la hora de crear el archivo *Excel* a descargar, había que programarlo como si fuera un archivo *.xml*. Es entonces cuando hay que especificar que se coja toda la tabla como si fuera un texto. Así, al pasarlo al *Excel*, que realmente es un *xml*, toda la tabla se cogía bien y no hubo problemas con los caracteres especiales (acentos, eñes...). También hay que tener en cuenta que este problema se ha replicado para cada uno de los formatos que se soporta para hacer la descarga. Con lo que, en cada uno de esos casos, ha sido necesario tener que definir en *xml* o en la función correspondiente cómo se tenía que introducir el texto para que la tabla se viera tal cual aparecía en el sitio web.

Independientemente del navegador desde el cual se haya realizado la descarga, al abrir el archivo *Excel* aparecerá una advertencia. Esta advertencia no supone ningún peligro a la hora de abrir el archivo, simplemente se reconoce que el archivo *Excel* que se está intentando abrir no es realmente *Excel*, sino que el código xml generado por las tablas y los parámetros necesarios para crear el archivo *Excel* se están haciendo pasar por un archivo *Excel*. Una vez pasar esa advertencia se verá que, en efecto, el archivo se ha descargado de manera satisfactoria.

## **8.2 Cambios en la base de datos**

La base de datos prácticamente no ha sufrido cambios desde que se cogió el proyecto. Esto se debe a que en general la estructura es buena y las funcionalidades que se han tenido que implementar no han requerido hacer cambios en la base de datos. En la siguiente imagen se ve el modelo de dominio final.

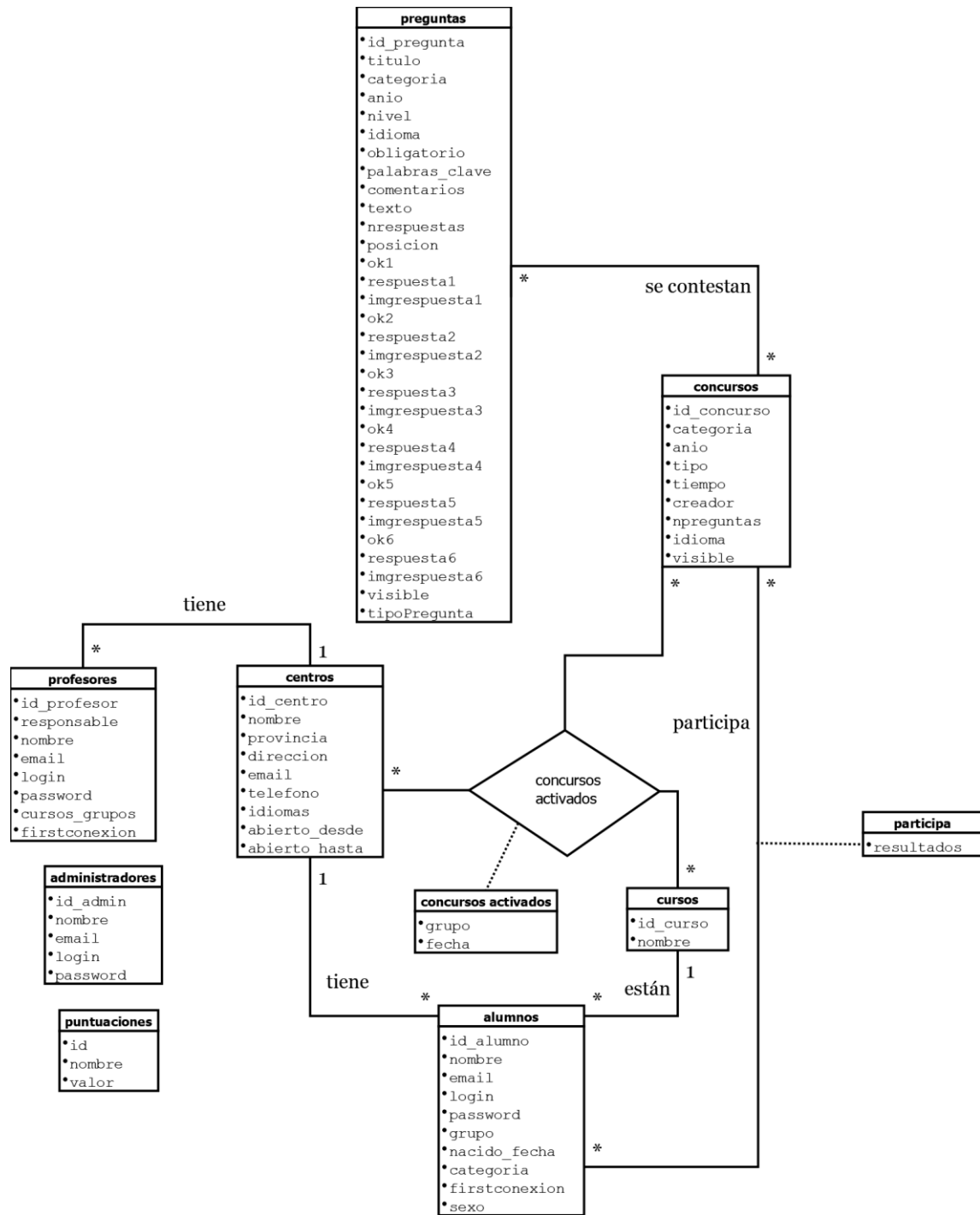


Figura 28. Modelo de dominio final

Como se puede apreciar en la Figura 23, todas las tablas y relaciones son exactamente iguales que las descritas en el apartado de modelo de dominio (Sección 5.3). La única excepción es la entidad registros.

Este cambio ha sido necesario por la automatización de los registros de los usuarios en el sitio web. Ahora ya no hace falta almacenar esos registros porque tan pronto como se finaliza el registro, se crea el centro, se busca y después al responsable de ese centro se le asigna el identificador del centro, para así saber de qué colegio es responsable ese profesor. Con respecto a los mensajes que se le envían tanto al profesor que se ha registrado como a *Bebras*, no es necesario almacenar nada al ser simplemente un mensaje informativo del registro.

## 9. Verificación y evaluación

En este apartado se van a exponer las pruebas que se han hecho de las funcionalidades completadas. Dada la naturaleza de este trabajo y teniendo en cuenta los lenguajes de programación en los que se ha trabajado, no hay una manera de automatizar las pruebas ni de hacer pruebas como tal. Por lo tanto, se van a documentar las pruebas que se han tenido que hacer a prueba y error, hasta dar con el resultado deseado. Antes de subir los cambios al servidor, obviamente todas estas pruebas se han realizado a nivel local.

Las pruebas que se han hecho son las siguientes:

<b>Registro</b>
<b>Descripción de la prueba:</b> El usuario no identificado se puede registrar en el sistema una vez proporcionar los datos exigidos.
<b>Salida esperada:</b> Los datos con los que se ha registrado aparecen en pantalla. El usuario recibe un correo a la dirección de correo electrónico que ha proporcionado con esos datos y un usuario y contraseña para acceder al sitio web.
<b>Resultado obtenido:</b> Funciona.

<b>Captcha</b>
<b>Descripción de la prueba:</b> Antes de finalizar el registro el usuario no identificado debe completar el registro completando el <i>captcha</i> . Solo hace falta indicar el valor resultado de la operación que aparece justo encima del campo en blanco.
<b>Salida esperada:</b> Si el campo se ha rellenado correctamente el registro finalizará de manera satisfactoria. Si el resultado introducido es incorrecto, aparecerá un <i>pop-up</i> indicando que la respuesta es incorrecta e impide seguir adelante con el registro mientras la respuesta no sea correcta.
<b>Resultado obtenido:</b> Funciona.

<b>Comprobar colegio en provincia al registrar</b>
<b>Descripción de la prueba:</b> Durante el registro se comprueba si el nombre del colegio introducido ya está en la provincia que se ha seleccionado una vez todo lo demás esté correcto.
<b>Salida esperada:</b> Si el colegio no está en esa provincia se permite el registro. Si ese no es el caso, no se permite el registro porque ya hay un colegio en esa provincia, con lo que el profesor tendrá que contactar con <i>Bebras</i> para ver si esto es correcto o no.
<b>Resultado obtenido:</b> Funciona.

<b>Estadísticas</b>
<b>Descripción de la prueba:</b> Un usuario identificado en el sitio web puede ver las distintas estadísticas que se ofrecen respecto a los concursos que se han realizado.
<b>Salida esperada:</b> Las estadísticas muestran la información deseada correctamente y cambian dependiendo de los filtros que se hayan seleccionado.
<b>Resultado obtenido:</b> Funciona.

<b>Estadísticas tablas</b>
<b>Descripción de la prueba:</b> Un usuario identificado en el sitio web puede ver las distintas estadísticas en las tablas que se ofrecen respecto a los concursos que se han realizado. Además, se permite ver si las respuestas que los alumnos han dado a las preguntas son correctas o no.
<b>Salida esperada:</b> Las estadísticas muestran la información deseada correctamente. El fondo se colorea de verde si la respuesta es correcta y se colorea de gris si la pregunta no se ha respondido.
<b>Resultado obtenido:</b> Funciona.

<b>Imágenes de preguntas interactivas no se visualizan bien</b>
<b>Descripción de la prueba:</b> A la hora de realizar un concurso se ven en pantalla las distintas preguntas que conforman el concurso que se va a hacer.
<b>Salida esperada:</b> Las preguntas se ven en el orden indicado y muestran la información correcta. El sitio web también identifica correctamente el tipo de pregunta que se está realizando (tipo test o interactiva).
<b>Resultado obtenido:</b> Funciona.

<b>Datos introducidos por el usuario se comprueban en la base de datos para evitar duplicados</b>
<b>Descripción de la prueba:</b> El usuario no identificado introduce los datos al registrarse.
<b>Salida esperada:</b> No hay problemas de comprobación que antes había entre los datos que había en la base de datos y los datos introducidos por el usuario por codificaciones diferentes. La comparación se hace correctamente y detecta bien todos los caracteres problemáticos (acentos, apóstrofes...).
<b>Resultado obtenido:</b> Funciona.

<b>Envío de correos a <i>Bebras</i> y usuario que se registra</b>
<b>Descripción de la prueba:</b> El usuario no identificado se puede registrar en el sistema una vez proporcionar los datos exigidos.
<b>Salida esperada:</b> Los datos con los que se ha registrado aparecen en pantalla. El usuario recibe un correo a la dirección de correo electrónico que ha proporcionado con esos datos y un



usuario y contraseña para acceder al sitio web.
---

<b>Resultado obtenido:</b> Funciona.
--------------------------------------

<b>Exportar tablas estadísticas</b>
-------------------------------------

<b>Descripción de la prueba:</b> Un usuario que se haya identificado en el sitio web puede exportar las tablas que estén disponibles en las distintas estadísticas.
---

<b>Salida esperada:</b> Aparece un pop-up informando que se va a hacer una descarga y se descarga un archivo <i>Excel</i> con los datos de la tabla.
--

<b>Resultado obtenido:</b> Funciona.
--------------------------------------

## 10. Conclusiones y trabajo futuro

Por último, se van a presentar las conclusiones que se han sacado del trabajo realizado. Para ello, se va a hacer un análisis entre la planificación que se estimó al iniciar el proyecto y la que al final ha sido necesaria para llevarlo a cabo.

Asimismo, también se va a hablar sobre algunas líneas de trabajo de cara al futuro junto a una estimación temporal que supondría realizarlas y una valoración de su utilidad, la licencia que se aplica al proyecto y una reflexión personal sobre el trabajo que se ha hecho.

### 10.1 Análisis entre planificación estimada y real

Considerando el planteamiento inicial que se ha hecho, ni los objetivos planteados, ni las herramientas utilizadas, ni la gestión de riesgos realizada han variado con respecto a lo que se esperaba. Al haber tenido que incorporar nuevas funcionalidades con respecto a lo esperado inicialmente, las herramientas utilizadas, el alcance, la planificación temporal y la evaluación económica previstos al inicio se han visto alterados.

#### 10.1.1 Herramientas utilizadas

Se ha utilizado una herramienta que no se previó utilizar al inicio del proyecto:

- **GanttProject:** Es un programa que permite crear diagramas Gantt de forma fácil y cómoda. Con solo tener creado un diagrama Gantt, este programa puede crear un diagrama PERT del proyecto que se esté realizando. Los diagramas Gantt que se hagan en este proyecto se han realizado con esta herramienta. Su página oficial es: <https://www.ganttproject.biz/>.

Dado que apenas se ha necesitado tiempo para aprender a utilizar esta herramienta, no se tendrá en cuenta en el alcance final o en el diagrama Gantt final.

### 10.1.2 Alcance

Además de las tareas explicadas en el planteamiento inicial (Sección 3.3.5), en el siguiente diagrama EDT de la Figura 29 se pueden ver las nuevas tareas que se han identificado en la implementación y desarrollo:



Figura 29. Diagrama EDT final de implementación y desarrollo

Las tareas nuevas son:

Captcha
<b>Bloque al que pertenece:</b> Implementación y desarrollo.
<b>Duración estimada:</b> 8 horas.
<b>Descripción de la tarea:</b> Introducir un <i>captcha</i> cuando un usuario se quiera registrar en el sistema para evitar que aplicaciones que pueden repetir esta tarea de manera automática saturen la base de datos y el sitio web.
<b>Salidas/Entregables:</b> Fichero PHP con el código que permite ejecutar correctamente esta tarea.
<b>Recursos necesarios:</b> Sublime Text 3, Apache, FileZilla, MySQL, Conexión a Escritorio Remoto y ordenador con conexión a Internet.

Exportar tablas estadísticas
<b>Bloque al que pertenece:</b> Implementación y desarrollo.
<b>Duración estimada:</b> 15 horas.
<b>Descripción de la tarea:</b> Posibilidad de poder exportar las tablas que se muestran como estadísticas de los concursos que los alumnos hayan realizado.
<b>Salidas/Entregables:</b> Fichero PHP con el código que permite ejecutar correctamente esta tarea.
<b>Recursos necesarios:</b> Sublime Text 3, Apache, FileZilla, MySQL, Conexión a Escritorio Remoto y ordenador con conexión a Internet.

Sanear los datos de la base de datos y sitio web
<b>Bloque al que pertenece:</b> Implementación y desarrollo.
<b>Duración estimada:</b> 10 horas.
<b>Descripción de la tarea:</b> Limpiar los datos que se encuentran en las base de datos. Además, hay que introducir las cadenas de caracteres tal cual se introducen en los cuadros de texto. Al sacar los datos de la base de datos también se tienen que conseguir correctamente, con la codificación adecuada.
<b>Salidas/Entregables:</b> Fichero PHP con el código que permite ejecutar correctamente esta tarea.
<b>Recursos necesarios:</b> Sublime Text 3, Apache, FileZilla, MySQL, Conexión a Escritorio Remoto y ordenador con conexión a Internet.

Tener nuevas tareas que en un principio no se habían previsto ha supuesto tener que dedicarle más tiempo del que se había esperado a las tareas predecesoras a estas nuevas tareas. La siguiente tabla muestra una comparativa de las horas que se estimaron en un inicio con las horas que realmente se ha necesitado. En esta tabla se han incluido las nuevas tareas.

ID	Nombre de la tarea	Predecesores	Horas estimadas	Horas reales
	<b>Aprendizaje</b>		<b>65</b>	<b>103</b>
1	Sublime Text 3	-	5	15
2	XAMPP	-	5	5
3	JavaScript	-	15	23
4	PHP	-	25	35
5	HTML	-	15	25
	<b>Organización</b>		<b>27</b>	<b>28</b>
6	Planificación de tareas a realizar	-	5	10
7	Selección de software	6	5	5
8	Instalación de software	7	7	3
9	Reuniones con los directores del proyecto	-	10	10
	<b>Captura de requisitos</b>		<b>15</b>	<b>20</b>
10	Casos de uso	6	10	10
11	Modelo de dominio	6, 10	5	10
	<b>Análisis y diseño</b>		<b>22</b>	<b>19</b>
12	Diagrama de clases	11	7	11
13	Diagramas de secuencia	12	15	8
	<b>Implementación y desarrollo</b>		<b>30</b>	<b>90</b>
14	Registro	1, 2, 3, 4, 5, 10, 11, 12, 13	15	23
15	Estadísticas	1, 2, 3, 4, 5	15	34
16	Captcha	1, 2, 3, 4, 5	-	8
17	Exportar tablas estadísticas	1, 2, 3, 4, 5	-	15
18	Sanear datos de la base de datos y sitio web	2, 4	-	10
	<b>Pruebas</b>		<b>35</b>	<b>53</b>
19	Pruebas de las funcionalidades	14, 15, 16, 17, 18	30	50
20	Evaluación del sitio web	19	5	3
	<b>Documentación</b>		<b>113</b>	<b>126</b>
21	Comentar el código	14, 15, 16, 17, 18	15	10
22	Material de apoyo	6	10	20
23	Diagramas y tablas	6	8	6
24	Memoria	7, 8, 21, 22, 23	60	75
25	Defensa del proyecto	24	20	15
	<b>TOTAL</b>	<b>24</b>	<b>307</b>	<b>439</b>

Tabla 3. Planificación temporal final

### 10.1.3 Planificación temporal

Con respecto a lo planteado al inicio se han tenido que hacer algunos cambios a la hora de dedicar tiempo a las distintas tareas. Como consecuencia, se ha necesitado más tiempo en las tareas relacionadas con la organización e implementación de las funcionalidades nuevas.

Los cambios más importantes han ocurrido en el aprendizaje, las estadísticas y la memoria. En un principio no se esperaba tener que trabajar tanto con las estadísticas que se pueden ver de los concursos. Como consecuencia, el tiempo que se ha necesitado para realizar la funcionalidad ha aumentado, además de tener que haber investigado un poco más de PHP para poder terminar la funcionalidad y de haber probado que mostrara los datos que se querían mostrar.

Dados estos cambios, el diagrama Gantt final resultante es el que aparece en la Figura 30. En este diagrama se puede ver la manera en la que las nuevas e imprevistas tareas han impactado sus predecesoras, siendo necesario aún más tiempo para terminarlas.

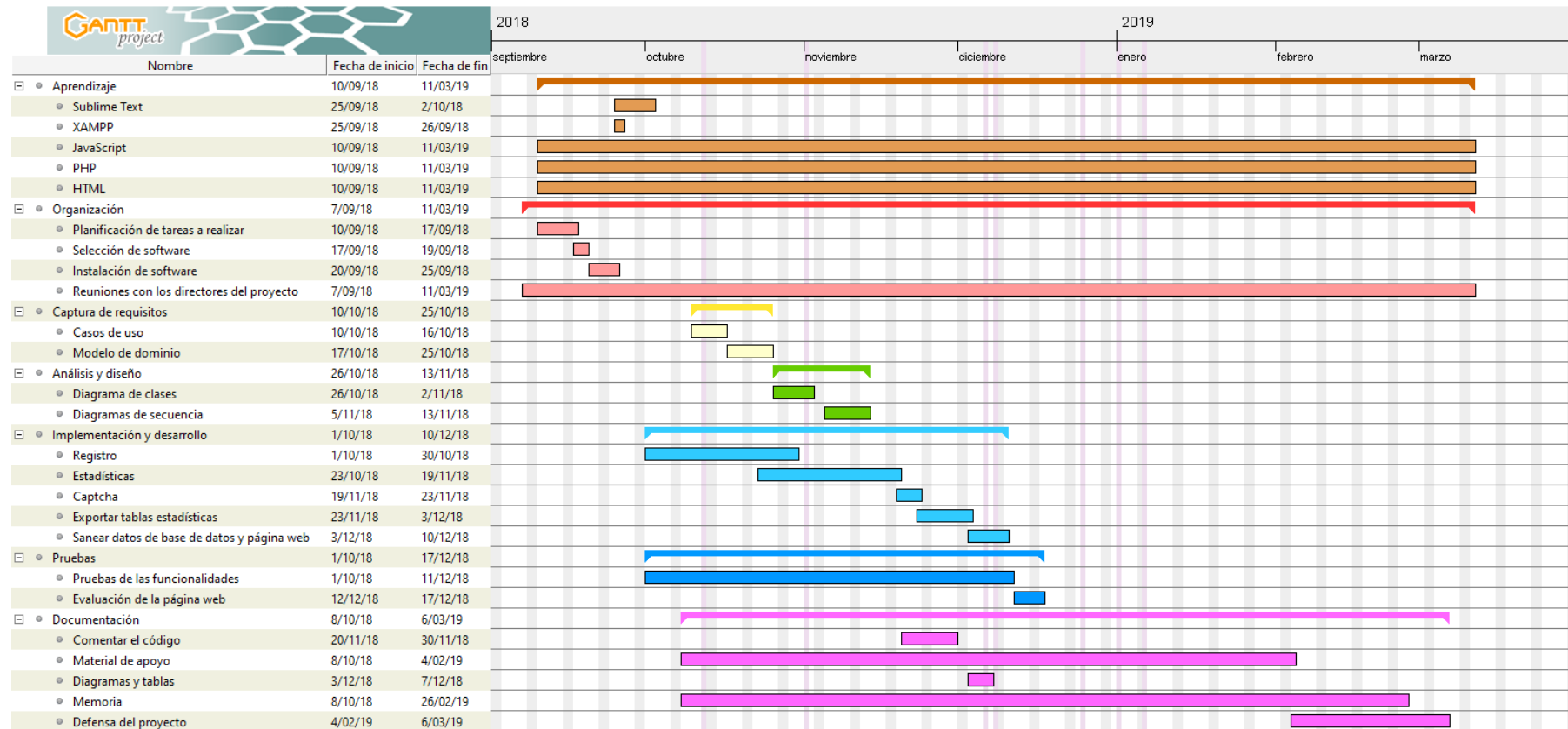


Figura 30. Diagrama de Gantt final

### 10.1.4 Evaluación económica

Los nuevos cálculos temporales han hecho que la evaluación económica que se había previsto al inicio también quedara obsoleta.

Recordemos que el sueldo por hora previsto no varía, con lo que sigue siendo de 7,70 € por cada hora trabajada.

Al final se han necesitado unas 439 horas para finalizar el proyecto. Sin embargo, de todas esas horas, 103 de ellas se van a emplear para aprender a manejar las herramientas y lenguajes de programación que se van a utilizar en el proyecto. Por lo tanto, se descuentan:

$$\text{Horas totales} - \text{Horas de aprendizaje} = 439 - 103 = 336 \text{ horas}$$

Por lo tanto, para saber el gasto de obra hay que multiplicar las horas que se trabajarán por lo que se pagará por hora:

$$\text{Horas trabajo} * \text{Sueldo/hora} = 336 * 7,70 = 2587,2 \text{ €}$$

Con lo que el gasto económico por parte de la mano de obra es de 2587,2 €.

Las reuniones con los directores del proyecto han terminado teniendo una duración de unas 10 horas, con lo que el coste no varía con lo que se expuso en la Sección 3.6.1. Así, la mano de obra asciende a 2687,2 €.

En la Tabla 4 se muestran los gastos finales del proyecto. Solo ha cambiado el coste de la mano de obra, con lo que el resto permanece igual.

Tipo de gasto	Coste en euros
Mano de obra	2687,2 €
Gastos software	36,60 €
Gastos hardware	246,5 €
Gastos indirectos	250 €
<b>TOTAL</b>	<b>3220,3 €</b>

Tabla 4. Gastos totales finales del proyecto



## 10.2 Líneas futuras

Este proyecto es muy extenso y, en función de las necesidades que puedan surgir, se puede seguir extendiendo para poder ofrecer la mejor experiencia a los usuarios que la utilicen. Es por eso que se han identificado algunas de las funcionalidades que se podrían realizar en el caso de que alguien cogiera este proyecto:

- ❖ **Juntar el foro con el sitio web:** Ahora mismo la cuenta que hay para entrar al sitio de *Bebras* y la que se utiliza para acceder al foro son distintas. Por ello, podría ser interesante poner en común estas cuentas, para que con una única cuenta se pudiera tener acceso a ambos sitios. En el caso de que el usuario no quisiera tener una misma cuenta asignada a ambos sitios, se le podría preguntar si quiere utilizar un mismo correo electrónico o una misma cuenta para el sitio web y el foro.
- ❖ **Ajustar el sitio web para dispositivos móviles y tabletas:** Normalmente se espera que a la hora de afrontar el desafío de *Bebras* los alumnos se conecten con los ordenadores que hay en sus escuelas. Sin embargo, sería bastante cómodo poder adaptar este desafío a otro tipo de dispositivos. Por ejemplo, móviles, tabletas, macs... para que así los alumnos pudieran realizar los ejercicios en cualquier momento. La única condición para poder realizar los ejercicios sería tener una cuenta en el sitio web.
- ❖ **Actualizar la situación de los alumnos:** Hay que tener en cuenta que cuando los alumnos se dan de alta en el sistema lo hacen en una categoría concreta. El problema surge cuando los alumnos, como norma general, avanzan cursos y adquieren conocimiento. Por lo tanto, los datos que se tienen de ese alumno tendrían que adaptarse al nivel académico que este tenga en un momento determinado. También habría que tener en cuenta a los alumnos que repitieran curso, porque estos no podrían avanzar a categorías posteriores.
- ❖ **Mostrar estadísticas sobre concursos no oficiales:** Las estadísticas que se muestran actualmente se corresponden con los concursos oficiales, que son los que los administradores del sitio web crean. Los profesores, tanto los responsables como los que no, pueden también crear concursos, aunque no se muestran estadísticas de los mismos. Por lo tanto, sería interesante que se pudieran ver las estadísticas de estos concursos, así como ver qué preguntas escogen para organizarlos.
- ❖ **Permitir a los profesores crear sus propias preguntas:** Tal y como están ahora las cosas, para poder crear un concurso los profesores tienen que escoger las

preguntas que los administradores hayan introducido en el sistema. Sería interesante que los profesores pudieran crear sus propias preguntas, siguiendo algunas pautas para poder introducirlas correctamente en la base de datos.

- ❖ **Filtros para los concursos creados por los profesores:** Ahora mismo se muestran todos los concursos que hay en la base de datos. Dado que no tienen por qué recordar todos los concursos que han hecho podría ser de utilidad poner un filtro.

## 10.3 Licencias

En este apartado se verán las licencias con las que se han trabajado en este proyecto. Hay que tener en cuenta la licencia bajo la que están las preguntas y la licencia bajo la que está el trabajo desarrollado.

### 10.3.1 Programas

De los programas que se han utilizado (Sección 3.2) su uso es completamente gratuito. Como en ningún caso ha sido necesario modificar o distribuir alguna de estas herramientas no se han comprobado a fondo las licencias de esos programas. Con saber que son de uso gratuito ha sido suficiente.

### 10.3.2 Preguntas

En cuanto a las preguntas se refiere, la licencia bajo la que están es la licencia de *Creative Commons - Reconocimiento-NoComercial-CompartirIgual (CC BY-NC-SA)*. [17] Esta licencia permite:

- Compartir: Copiar y redistribuir el material en cualquier medio o formato.
- Adaptar: Entremezclar, transformar y crear a partir del material.

## 10.4 Reflexión personal

Estoy contento con las cosas que he podido aprender de este proyecto. En general, no sabía que esperar del proyecto ya que es el primero que hago y tampoco

sabía que iba a involucrar tener que hacer uno. Me ha resultado bastante interesante hacerlo porque se asemeja mucho a las funcionalidades que un cliente puede pedirle a una empresa, con lo que es como estar trabajando en una empresa que se encarga de este tipo de cosas.

También estoy satisfecho con haber salido de mi zona de confort, ya que he podido ver qué tipo de sintaxis utilizan los lenguajes de programación en los que he trabajado. En este aspecto, he visto la utilidad de varias asignaturas que he cursado a lo largo de la carrera. Gracias a ellas, he tenido esas nociones básicas de programación que me han sido útiles a la hora de enfrentar los distintos retos que han supuesto las funcionalidades realizadas. Y, si bien es cierto que lo que he visto y hecho es solo una base, pienso que es bueno disponer de una a directamente no tener esa base que he obtenido habiendo trabajado en estos lenguajes de programación.

Un aspecto muy interesante de este trabajo es el hecho de que el sitio web está operativo y hay usuarios que la están utilizando. Esto resulta motivador, ya que se sale de la rutina escolar de tener que hacer ejercicios para entregarlos y nada más. Bien es cierto que de esos ejercicios hechos se obtiene conocimiento, aunque es un tanto desalentador que no terminen siendo nada más.

Por último, me gustaría dar las gracias a Eugenio Bravo y Javier Bilbao por haber propuesto este trabajo. Ha sido una experiencia enriquecedora y me ha servido para ver qué es lo que hay detrás de un sitio web. Su ayuda a la hora de hacer modificaciones y saber dónde estaban los fallos que había que arreglar ha sido útil a la hora de enfrentar a estos problemas.

## 11. Bibliografía

- [1] Apache Friends. ¿Qué es XAMPP? <https://www.apachefriends.org/es/index.html>  
Accedido: 21-01-2019.
- [2] Anders Borg. What is a bot? <https://www.quora.com/What-is-a-bot>, 2016.  
Accedido: 21-01-2019.
- [3] Computer hoy. Qué son los captchas y cómo funcionan.  
<https://computerhoy.com/noticias/software/que-son-captchas-como-funcionan-10507> Accedido 20-01-2019.
- [4] GlobalSign. What is an SSL Certificate? <https://www.globalsign.com/en-ph/ssl-information-center/what-is-an-ssl-certificate/> Accedido: 21-01-2019.
- [5] DataTables. Add advanced interaction controls to your HTML tables the free & easy way. <https://datatables.net/> Accedido: 21-01-2019.
- [6] FileZilla. What is FTP? <https://filezilla-project.org/faq.php>. Accedido 21-01-2019.
- [7] Java. How is JavaScript different from Java?  
[https://java.com/en/download/faq/java\\_javascript.xml](https://java.com/en/download/faq/java_javascript.xml) Accedido: 28-01-2019.
- [8] PHP. What can PHP do? <http://php.net/manual/en/intro-whatcando.php> Accedido: 28-01-2019.
- [9] Qloudea. Soporte. ¿Qué es TCP/IP?  
<https://soporte.qloudea.com/hc/es/articles/115003659065--Qu%C3%A9-es-TCP-IP->, 2017. Accedido: 21-01-2019
- [10] Techopedia. What is Widget?  
<https://www.techopedia.com/definition/3452/widget> Accedido: 21-01-2019.
- [11] Voogt, J., Knezek, G., Christensen, R., & Lai, K. W. (2018). Information and Communication Technology and Education: Meaningful Change through Teacher Agency. Accedido 25-12-2018.
- [12] Bebras. International Challenge on Informatics and Computational Thinking.  
<https://www.bebas.org/>. Accedido 15-01-2019.
- [13] Boletín Oficial del Estado. <https://www.boe.es/boe/dias/2018/03/06/pdfs/BOE-A-2018-3156.pdf>. Accedido 16-01-2019.
- [14] Sabadías, A. V. (1995). Estadística descriptiva e inferencial (Vol. 8). Univ de Castilla La Mancha. Accedido 20-01-2019.

[15] García, H., & Matus, J. (2003). Estadística descriptiva e inferencial I. Accedido 20-01-2019.

[16] Sivakorn, S., Polakis, J., & Keromytis, A. D. (2016). I'm not a human: Breaking the Google reCAPTCHA. Black Hat. Accedido 20-12-2018.

[17] Creative Commons. Reconocimiento-NoComercial-CompartirIgual 3.0 España (CC BY-NC-SA 3.0 ES). <https://creativecommons.org/licenses/by-nc-sa/3.0/es/> Accedido 20-01-2019.