

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



TRABAJO FIN DE MÁSTER

Curso online de C interactivo

Máster en Ingeniería Informática

Autor: Hernández Molpeceres, Andrés

Tutor: Sierra Urrecho, Alejandro

Septiembre, 2016

Resumen

En el presente trabajo se tratará el desarrollo de una aplicación web, enfocada al aprendizaje del lenguaje de programación C, ya que éste es uno de los principales lenguajes utilizados para los cursos de introducción a la programación. El propósito final de la aplicación, es que sea utilizada por los alumnos de primer año de la Escuela Politécnica Superior como herramienta complementaria y de apoyo a la docencia.

Para el desarrollo del software de la aplicación, se ha llevado a cabo un desarrollo clásico en cascada. En primer lugar, se ha estudiado el mercado de aplicaciones especializadas en el aprendizaje de lenguajes de programación, para obtener referencias y poder realizar un desarrollo con perspectiva. En la fase de diseño, se ha establecido la arquitectura que implementará la aplicación tanto a nivel de cliente como de servidor, las tecnologías para su desarrollo y el contenido de la misma.

Una vez definido el diseño, se ha pasado a desarrollar la aplicación web según éste mismo. Para ello, mediante el uso de las tecnologías y herramientas elegidas, como el framework PHP Laravel, se han resuelto las diversas dificultades técnicas a las que se enfrentaba este proyecto, como han sido la compilación del código en el servidor, ejecución con inputs de usuario, generación de gráficas o integración de los distintos elementos en la interfaz de usuario, entre otras.

Finalmente se han realizado pruebas a la aplicación web ya finalizada y con sus requisitos completamente implementados. También se han llevado a cabo test de estrés al servidor para comprobar su capacidad. Después de la interpretación y análisis de los resultados, la aplicación desarrollada cumple con las expectativas fijadas y debería ser capaz de funcionar en un entorno de uso real.

Palabras clave: PHP, Laravel, aplicación web, e-learning, lenguaje de programación C

Summary

The present work will deal with the development of a web application. It will be focused on learning the C programming language, because this is one of the major languages used for initial programming courses. The final purpose of the application is that it is used by the first year students of the Higher Polytechnic School and as a complementary tool to support teaching.

For the development of application software, it has used a classical cascade software development. First, the market of specialized applications in learning programming languages has been studied, in order to get references and be able to make a development with a good perspective. In the design phase, the application architecture, at client and server side, the technologies for development and the content of it have been established.

Once the design of the application has been defined, it is time to begin with the implementation of it. To do this, using the technologies and tools chosen, like the PHP framework Laravel, various technical difficulties of the project have been solved, as compiling the code on the server, running a program with inputs from the user, graphics generation or the integration of the user interface elements, among others.

Finally, the completed web application, with all of its requirements implemented, has been tested. Also, stress tests have been conducted to check the server capacity. After interpretation and analysis of the results, the developed application meets the expectations and should be able to run in a real environment.

Keywords: PHP, Laravel, web application, e-learning, C programming language

Tabla de contenido

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
1.3	Estructura de la memoria.....	3
2	Estado del arte	4
2.1	Introducción al estado del arte	4
2.2	Principales plataformas.....	6
2.2.1	Cursos de lenguaje C	8
3	Análisis de requisitos.....	10
3.1	Funcionales.....	10
3.1.1	Requisito del módulo docente	10
3.1.2	Requisito del módulo de evaluación	12
3.2	No funcionales.....	12
4	Diseño.....	15
4.1	Arquitectura de la aplicación	15
4.1.1	Front-end.....	16
4.1.2	Back-end.....	21
4.2	Laravel framework	26
4.2.1	Características generales.....	26
4.2.2	Patrón de diseño software	27
4.2.3	Routing	28
4.2.4	Plantillas Blade	29
4.2.5	Middleware	31
4.2.6	Migraciones.....	31
4.2.7	Protección CSRF.....	32

4.3	Base de datos	33
4.4	Contenido docente.....	35
5	Implementación	36
5.1	Estructura de la aplicación	36
5.2	Módulo docente.....	37
5.2.1	Compilación, ejecución e interacción con el código c.....	38
5.2.2	Contenido docente.....	42
5.3	Módulo de evaluación.....	45
5.3.1	Librería gráfica.....	46
5.3.2	Pantalla de estadísticas	46
5.3.3	Pantalla de usuarios	49
5.3.4	Pantalla de bugs	50
6	Pruebas y resultados	51
6.1	Validación de requisitos	51
6.2	Pruebas del servidor web.....	53
6.2.1	Entorno de pruebas.....	54
6.2.2	Ejecución de las pruebas y análisis de resultados.....	55
7	Conclusiones.....	61
8	Referencias.....	62
9	Anexo A – Índice del contenido docente	64

Índice de tablas

Tabla 1 - Comparativa Mooc y Curso online	5
Tabla 2 - Comparativa edX - Coursera.....	7
Tabla 3 - Validación de requisitos	53
Tabla 4 - Características del servidor web.....	54
Tabla 5 - Resultados de la prueba de estrés	58

Índice de figuras

Figura 1 - Distribución de cursos por plataforma [5]	6
Figura 2 - Web Learn-C.org	8
Figura 3 – Arquitectura Cliente-Servidor	15
Figura 4 – Logo HTML5.....	16
Figura 5 - Ejemplo de formulario	16
Figura 6 – Logo CSS3.....	18
Figura 7 - Logo Javascript	18
Figura 8 - Logo JQuery.....	19
Figura 9 - Logo Bootstrap	20
Figura 10 - Logo WampServer	21
Figura 11 - Logo Apache	22
Figura 12 - Utilización de servidores	23
Figura 13 - Logo MySQL.....	24
Figura 14 - Logo php.....	25
Figura 15 - Logo Laravel	26
Figura 16 - Patrón MVC	28

Figura 17 - Esquema de la base de datos	34
Figura 18 – Interfaz para la programación	38
Figura 19 - Ejemplo de compilación	40
Figura 20 - Ejemplo de ejecución	41
Figura 21 - Pantalla completa del módulo docente	43
Figura 22 - Ejemplo de solución a ejercicio	44
Figura 23 - Pantalla de ayuda	45
Figura 25 - Progreso de los alumnos	47
Figura 26 - Estadísticas de usuario por unidad	47
Figura 27 - Tiempo por unidad	48
Figura 28 - Pantalla de usuarios	49
Figura 29 - Pantalla de bugs	50
Figura 30 - Logo Apache JMeter.....	54
Figura 31 - Interfaz de JMeter	56
Figura 32 - Árbol de resultados de JMeter	57
Figura 33 - Error en petición HTTP	59
Figura 34 - Resultados de la prueba con 500 usuarios.....	60

1 Introducción

El aprendizaje de la programación tiene muchos puntos en común con el aprendizaje de idiomas extranjeros. A pesar de las importantes similitudes, es sorprendente que la enseñanza online de idiomas esté mucho más desarrollada que la de los lenguajes de programación, especialmente teniendo en cuenta que la programación es una disciplina informática. Si bien existen muchas herramientas informáticas para la enseñanza de idiomas, solo recientemente se empiezan a popularizar las correspondientes para la enseñanza de la programación [1, 2, 3], la mayor parte de ellas orientadas al aprendizaje de lenguajes como HTML, CSS, Bootstrap, JavaScript, etc.

1.1 Motivación

Aunque en la UAM existen distintos entornos de aprendizaje online como Moodle, o el más reciente Open edX, ninguno de ellos está especialmente diseñado para la docencia de la programación. Como primer requisito, una herramienta orientada a la docencia de la programación debería incluir el compilador del lenguaje correspondiente, de tal forma que los estudiantes no se vean obligados a instalar entornos de programación adicionales, para que simplemente se puedan poner a practicar con el lenguaje desde el primer momento. En ocasiones, los estudiantes de primeros cursos de programación encuentran dificultades al dar estos primeros pasos, lo cual retrasa considerablemente el proceso de aprendizaje.

Además, una herramienta específica permitiría, por ejemplo, corregir automáticamente el código generado por el estudiante durante su aprendizaje, a la vez que recogería la información generada durante todo el proceso tal y como los errores de compilación cometidos, el tiempo invertido en cada ejercicio, etc. Las herramientas tradicionales, aunque son susceptibles de recoger estos datos, no resultan muy amigables a la hora de presentarlos al profesor, lo que dificulta extraer conocimiento del proceso de aprendizaje que es, sin duda, una de las tareas pendientes de este tipo de plataformas.

Por estas razones, surge la necesidad de desarrollar una herramienta específica para el aprendizaje online del lenguaje de programación C [4]. Éste es el lenguaje que se estudia en los cursos de introducción a la programación de la EPS y, de acuerdo con nuestra información, no existe ningún tutor online específico para este lenguaje. Por lo tanto, este trabajo tratará sobre el desarrollo desde cero de una herramienta online de aprendizaje.

1.2 Objetivos

El objetivo principal de este TFM es el desarrollo de un curso online interactivo de C. Uno de los requisitos fundamentales de esta herramienta es liberar al usuario de la necesidad de instalar herramientas adicionales para el aprendizaje del lenguaje de programación. Por esta razón, será el servidor quien se encargará de compilar los ejercicios desarrollados por el estudiante.

El curso constará de dos módulos principales: el módulo docente y el módulo de evaluación. El módulo principal del curso es el módulo docente, el encargado de presentar al estudiante los objetivos. La idea fundamental que nos guía al desarrollar los objetivos es la de aprender resolviendo problemas. Los objetivos se presentarán de forma muy gradual: pequeñas píldoras de información acompañadas de un pequeño ejercicio, todo esto visible conjuntamente y de forma atractiva dentro de una misma pantalla. Es necesario recordar que esta herramienta está pensada para el aprendizaje autónomo del lenguaje de programación.

El módulo docente tendrá un carácter secuencial, es decir, no se podrá acceder a un objetivo sin haber completado los anteriores. Sin embargo, sí será posible repasar objetivos ya alcanzados. Éste es un aspecto fundamental del curso que influye claramente en la elaboración del material docente pues cuando el estudiante alcanza un objetivo, podemos contar con que ha superado los anteriores.

Una de las ventajas de disponer de una herramienta como ésta es que se puede recoger información sobre el proceso de aprendizaje de cada estudiante. Tradicionalmente, los estudiantes utilizan entornos de programación independientes como Netbeans, Visual Studio o Eclipse. El profesor es consciente de los errores en los que suelen caer los estudiantes por las sesiones de laboratorio, pero no tiene información estadística detallada al respecto, información que puede ser muy valiosa a la hora de diseñar los contenidos de los cursos.

En su forma más elemental, esta información consistirá en el grado de completitud de los objetivos planteados. El profesor podrá visualizar esta información mediante el módulo de evaluación. No solo esto, sino que también se podrá estimar la dificultad de cada objetivo a través del tiempo medio invertido por los estudiantes, así como a través del número de intentos llevados a cabo. Toda esta información se recogerá en una base de datos cuyo diseño es otro de los aspectos fundamentales de este TFM, y que permitirá mejorar el curso entre ediciones.

1.3 Estructura de la memoria

Capítulo 1 – Introducción: Exposición de la motivación y objetivos que se pretenden alcanzar en el presente trabajo.

Capítulo 2 – Estado del arte: Se dará una visión de las principales plataformas de aprendizaje online, su evolución, características y funcionalidades.

Capítulo 3 – Análisis de requisitos: Se listarán y explicitarán todos aquellos requisitos que serán necesarios incluir en la aplicación, para ofrecer toda la funcionalidad requerida y necesaria para un curso online.

Capítulo 4 – Diseño: En esta sección se desarrollará la fase de diseño para la aplicación. Se definirá la arquitectura de la aplicación, las tecnologías que empleará y el contenido de la misma.

Capítulo 5 – Implementación: Se detallarán las soluciones alcanzadas, exponiendo los métodos y técnicas utilizadas. También se mostrará y explicará cada una de las partes de la aplicación.

Capítulo 6 – Pruebas y resultados: En esta sección se explicarán las pruebas a realizar y los resultados de las mismas.

Capítulo 7 – Conclusiones: Se resumirán los puntos más importantes, las impresiones generales del proyecto y su ámbito de estudio y el trabajo futuro.

2 Estado del arte

En este apartado se analizarán las distintas plataformas de aprendizaje online, o e-learning, que existen en la actualidad. Este estudio del estado actual de las tecnologías y plataformas de e-learning, permitirá elaborar un software con cierta perspectiva de lo que se espera encontrar un usuario. También facilitará su familiarización con la aplicación y que ésta sea de la mayor calidad posible.

2.1 Introducción al estado del arte

Para los profesionales docentes encargados de enseñar programación, puede llegar a resultar muy desalentador el hecho de observar cómo los alumnos no llegan a comprender los conceptos más básicos y van perdiendo la atención y el interés en general. Esta situación ha dado lugar, desde épocas antiguas, a muchos estudios e investigaciones sobre el tema, tratando de buscar soluciones o métodos para paliar las dificultades de los estudiantes.

En lo referente a cursos de aprendizaje online, en general nos podemos encontrar con dos tipos de cursos:

- **Mooc:** Mooc es el acrónimo de Massive Open Online Course. Como su propio nombre indica se trata de cursos online gratuitos y masivos. En estos cursos puede apuntarse un número ilimitado de estudiantes a través de sus respectivas webs. Uno de los potenciales de estos cursos es la compartición y apoyo al aprendizaje por parte de todos los participantes del curso, de forma que cuantos más haya, más y mejores respuestas encontraremos a dudas que puedan surgir.
- **Curso online:** Los cursos online tradicionales, normalmente están dirigidos a unos estudiantes en concreto y no están abiertos a todo aquel que desee utilizarlo. Estos cursos online suelen ser de pago. Los profesores normalmente se encuentran a disposición de los estudiantes para poder hacerles consultas, y la parte colaborativa entre los alumnos no suele ser común.

Ambos tienen muchos elementos en común, pero veamos una tabla resumen de las principales diferencias entre ambos:

Mooc	Curso online tradicional
Puede desarrollarse en múltiples plataformas y encontrar variaciones o funcionalidades significativas entre ellas.	Se desarrolla en una plataforma con unas funcionalidades y una estructura muy acotada.
Dirigido a un entorno y público muy abierto	Dirigido a un entorno cerrado.
Servicio gratuito	Normalmente es un servicio de pago
Número de participantes muy elevado	Número de participantes reducido
Comunidad como principal fuente de ayuda y soporte.	Profesor como principal fuente de ayuda y soporte.
Énfasis en el procedimiento de aprendizaje más que en la evaluación.	Más orientado a la evaluación y obtención de resultados.
Infraestructuras y coste del entorno y equipos elevado, para dar soporte a tantos usuarios.	Infraestructura y coste asociado más reducido.
Problemas de seguridad potencialmente mayores.	Seguridad más controlada.

Tabla 1 - Comparativa Mooc y Curso online

2.2 Principales plataformas

Centraremos el estudio en las principales plataformas gratuitas de aprendizaje online. Estas son con una amplia diferencia edX y Coursera dentro de los Moocs.

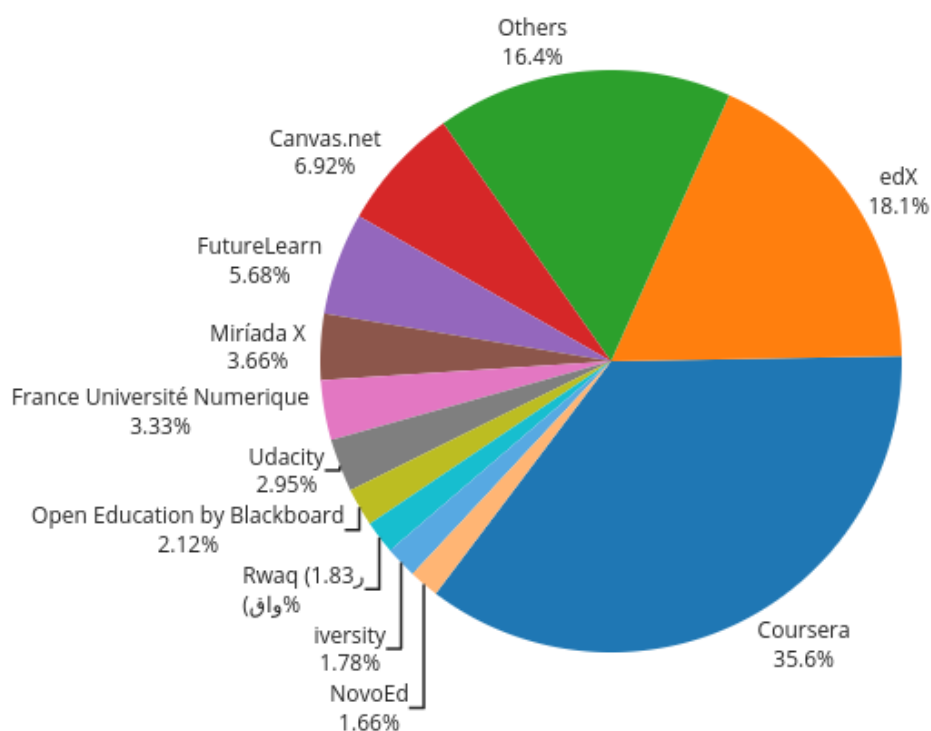


Figura 1 - Distribución de cursos por plataforma [5]

El análisis de cualquiera de estas plataformas puede llegar a ser muy extenso y complejo, por lo que para una mayor claridad y a modo de resumen, a continuación, se puede ver una tabla comparativa de ambas plataformas.

	edX	Coursera
Nº de cursos	Aproximadamente 950 en inglés y 88 en español.	Aproximadamente 1500 en inglés y 250 en español.

Comunidad	Los foros son mejorables, no incitan mucho a la participación.	Foros muy activos y con una gran comunidad detrás.
Tests y ejercicios	Cuenta con cuestionarios y ejercicios bastante complejos, los cuales realmente ponen a prueba nuestras capacidades.	En general son bastante abordables y permiten superar los cursos cómodamente si se han seguido correctamente.
Interfaz de usuario	Bastante completa a la vez que sencilla de aprender. Cuenta con ayudas de manejo y buenos menús de navegación.	Bastante similar a la de edX, también muy bueno, quizás algo menos atractiva.
Contenido didáctico	Muy buen contenido, muchas veces respaldado por importantes universidades.	Depende del curso, pero en general muy completo y avanzado.
Certificados	Certificados con una validez bastante apreciada, dada la rigurosidad de los cursos.	Certificados menos relevantes que los de edX.

Tabla 2 - Comparativa edX - Coursera

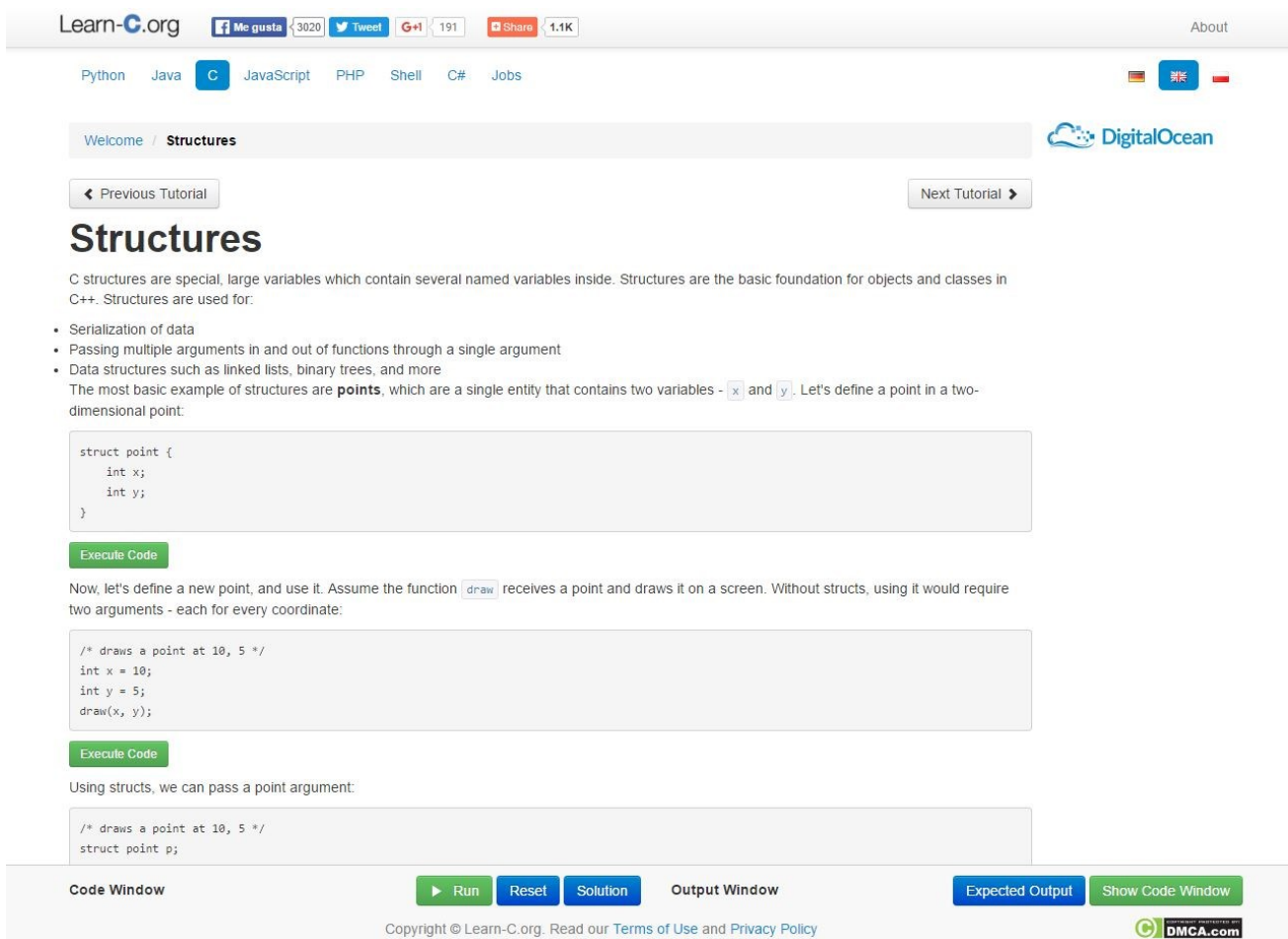
Dejando de lado los Moocs, las plataformas de aprendizaje online de lenguajes de programación más populares que podemos encontrarnos son Codeschool, Codecademy y W3schools. W3schools se centra sobre todo en lenguajes de desarrollo web como html, css, javascript, etc. Es una plataforma muy sencilla e intuitiva, cuenta con ejemplos prácticos y es una importante referencia para todos los desarrolladores web. Por otro lado, tenemos Codeschool y Codecademy, ambos muy parecidos en su manejo e interfaz, centrados principalmente en cursos de lenguajes para el desarrollo web como html, php, .net, ruby on rails, etc.

En general, un aspecto que destaca por su ausencia en la mayoría de plataformas y cursos online, es que están dirigidas exclusivamente a los estudiantes, es decir, un profesor no puede gestionar un grupo de alumnos y ver sus resultados. Estos datos suelen ser privados y las empresas que poseen las plataformas son las únicas que tienen acceso a ellos para su uso.

2.2.1 Cursos de lenguaje C

Si buscamos cursos online gratis específicos de lenguaje C, dentro de las plataformas más populares no los encontramos, o en todo caso breves introducciones a la programación en C sin apenas profundizar en él. Fuera de estas plataformas existen algunos cursos de C, la mayoría de pago. Muchos de ellos tampoco cuentan con un entorno de programación con consola de ejecución, se limitan simplemente a ir completando unos test o ejercicios.

Uno de los más interesantes es Learn-c [6]. Éste cuenta con varias unidades con contenido didáctico, un editor y una consola de ejecución. Como puntos flojos cuenta con una interfaz poco cómoda, bastante recargada de elementos y los contenidos son algo escasos.



The screenshot shows the Learn-C.org website interface. At the top, there's a navigation bar with social media links (Me gusta, Tweet, G+, 191, Share, 1.1K) and a language selector (About, Python, Java, C, JavaScript, PHP, Shell, C#, Jobs). The main content area is titled "Structures" and includes a "Welcome / Structures" breadcrumb, navigation buttons for "Previous Tutorial" and "Next Tutorial", and a list of topics: "Serialization of data", "Passing multiple arguments in and out of functions through a single argument", and "Data structures such as linked lists, binary trees, and more". The text explains that structures are used for data and provides an example of a 2D point structure. It shows two code snippets: one defining a point structure and another using it to draw a point. The interface includes a "Code Window" at the bottom with buttons for "Run", "Reset", "Solution", "Output Window", "Expected Output", and "Show Code Window".

```
struct point {
    int x;
    int y;
}
```

```
/* draws a point at 10, 5 */
int x = 10;
int y = 5;
draw(x, y);
```

```
/* draws a point at 10, 5 */
struct point p;
```

Figura 2 - Web Learn-C.org

La web ALISON [7], también ofrece un curso de C. Su contenido se queda escaso, llegando su contenido final a los bucles y funciones personalizadas. Tampoco cuenta con una interfaz de programación. La plataforma Udemy [8] ofrece un curso de C, pero tiene un precio de 19€ y tampoco especifica su alcance y contenido. Por último, existen múltiples páginas como

Cprogramming [9], las cuales consisten simplemente en lecciones para leer sin ejercicios ni entorno de programación alguno.

Como vemos en lo referente a cursos de C, las plataformas presentan muchas carencias, las cuales se tratarán de solventar en este trabajo, ofreciendo un entorno de aprendizaje lo más completo posible, con un contenido suficientemente amplio, ejercicios y un entorno de programación potente y fácil de utilizar.

3 Análisis de requisitos

En este apartado se buscará el definir las funcionalidades y comportamientos que deberá tener la aplicación. Mediante la división de la funcionalidad de la aplicación en tareas y funcionalidades, obtenemos una lista de requisitos básicos que ésta deberá cumplir. De esta forma, una vez terminado el desarrollo del sistema, se puede comprobar su validez y completitud evaluando si se cumplen los requisitos definidos.

3.1 Funcionales

Los requisitos funcionales definen aquellas funciones que un software debe cumplir y que definen el comportamiento del sistema. Ya que la aplicación va a contar con dos módulos bien definidos, dividiremos los requisitos entre ambos para una mayor claridad.

3.1.1 Requisito del módulo docente

- RF-1

La aplicación deberá contar con funcionalidad para el registro de nuevos usuarios mediante un formulario web.

- RF-2

La aplicación deberá contar con funcionalidad para el control de acceso a la aplicación mediante el login los usuarios previamente registrados.

- RF-3

La aplicación deberá guardar el progreso de los usuarios según avancen en el curso.

- RF-4

La aplicación deberá presentar el contenido docente relativo a cada capítulo y permitir la navegación entre ellos.

- RF-5

La aplicación deberá mostrar ejercicios a resolver para los usuarios, comprobará su corrección e informará de la solución propuesta cuando proceda.

- RF-6

La aplicación podrá evaluar, mediante tests, el contenido aprendido en cada sección.

- RF-7

La aplicación deberá controlar el avance de los usuarios, no pudiendo pasar a un siguiente capítulo sin haber superado su predecesor.

- RF-8

La aplicación deberá permitir consultar el contenido de capítulos superados.

- RF-9

La aplicación deberá almacenar en una base de datos los datos de los usuarios, su progreso, estadísticas y resultados de los ejercicios.

- RF-10

La aplicación deberá poder compilar código C, informando de posibles errores al usuario.

- RF-11

La aplicación deberá ejecutar código C escrito por el usuario.

- RF-12

La aplicación deberá permitir interactuar con el programa C mediante inputs del usuario y mostrar la salida correspondiente.

- RF-13

La aplicación deberá contar con un editor de texto en el cual los usuarios podrán desarrollar sus programas.

- RF-14

La aplicación deberá permitir el desarrollo de al menos una librería junto con el main.

- RF-15

La aplicación deberá contar con un sistema de reporte de bugs de la misma.

3.1.2 Requisito del módulo de evaluación

- RF-16

La aplicación deberá generar estadísticas de medidas como los tiempos medios por lección, número de intentos, resultados medios de los tests, etc.

- RF-17

La aplicación deberá permitir la administración de los usuarios registrados.

- RF-18

La aplicación deberá facilitar la gestión de los bugs reportados por los usuarios.

3.2 No funcionales

De seguridad:

- RNF-1

El servidor deberá controlar la ejecución de código peligroso o que pueda perjudicar el rendimiento del mismo. (bucles infinitos, llamadas al sistema, creación y borrado de ficheros, etc.)

- RNF-2

La aplicación controlará el acceso al módulo de evaluación, no siendo accesible para estudiantes.

- RNF-3

La aplicación solicitará una contraseña con al menos 6 caracteres de longitud a la hora de registrar nuevos usuarios.

- RNF-4

La aplicación evitará la formación de consultas susceptibles de sufrir inyección de código SQL.

De usabilidad:

- RNF-5

El editor de texto destinado a programar contará con ayudas a la programación, como autocompletado de paréntesis y llaves, autoindentación, asignación automática de colores a palabras clave, plegar y desplegar llaves, paréntesis, etc.

- RNF-6

El módulo de evaluación permitirá la visualización gráfica de las estadísticas de una forma sencilla y clara.

- RNF-7

En los ejercicios a resolver, se dispondrá de un botón ayuda que desplegará un pequeño resumen de la lección.

- RNF-8

La aplicación será adaptable. a distintas resoluciones y dispositivos.

De mantenibilidad:

- RNF-9

La aplicación deberá contar con un sistema que permita reportar bugs.

- RNF-10

La aplicación deberá permitir visualizar y actualizar el estado de los bugs reportados.

- RNF-11

La aplicación deberá contar con funcionalidad para recordar la contraseña en caso de que un usuario la olvidase.

De rendimiento:

- RNF-12

La aplicación deberá dar servicio a múltiples usuarios concurrentes.

- RNF-13

Los tiempos de respuesta del servidor no deberán ser superiores a 10 segundos.

4 Diseño

En este capítulo, después de haber establecido los requisitos y teniendo bien definida la funcionalidad de la aplicación, pasaremos a tratar la parte del diseño de la misma. Para ello, se definirá la arquitectura de la aplicación, las tecnologías que empleará y el contenido de la misma.

4.1 Arquitectura de la aplicación

Se usará un modelo Cliente-Servidor. Podemos distinguir dos elementos bien diferenciados en la arquitectura de la aplicación. Por un lado, tendremos el lado del cliente (Front-end), es decir, la parte con la que los usuarios llevarán a cabo la interacción con la aplicación. Por el otro se encontrará el lado del servidor (Back-end), encargado de ofrecer los servicios y dar respuesta a las peticiones de los clientes.

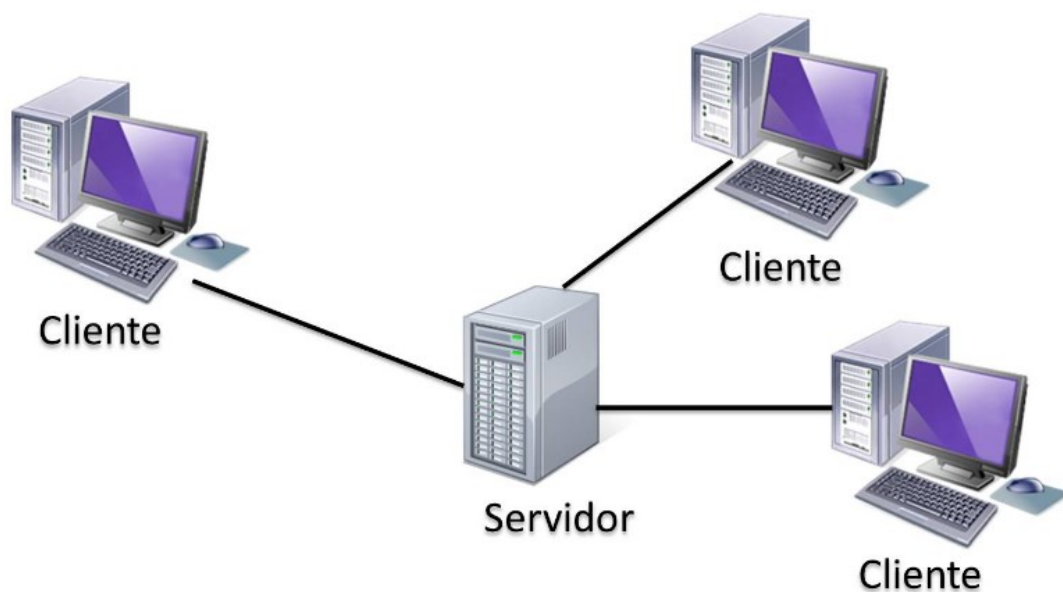


Figura 3 – Arquitectura Cliente-Servidor

4.1.1 Front-end

En esta parte de la aplicación los datos recibidos del servidor deben ser procesados e interpretados por el cliente. Con el fin de asegurar un diseño limpio, claro y mantenible, es muy recomendable dividir la parte del contenido, de la del diseño/estilo y la del procesado. Para cumplir con esta buena práctica en el proyecto se emplearán las siguientes tecnologías:

HTML



Figura 4 – Logo HTML5

HTML (*HyperText Markup Language*) es un lenguaje de marcado empleado para el desarrollo de páginas web. A través de este lenguaje creamos la estructura de nuestra página. Para ello existe una colección de tags definidos. Con estos tags se definen los distintos elementos que forman toda página web, como son formularios, listas, imágenes. Veamos un ejemplo de un formulario en html y su resultado en el navegador:

```
<form action="a">  
  Nombre:<br>  
  <input type="text" name="nombre">  
  <br>  
  Apellido:<br>  
  <input type="text" name="apellido">  
  <br><br>  
  <input type="submit" value="Submit">  
</form>
```

A rendered HTML form displayed in a browser. It consists of a vertical line on the left side. To the right of the line, the text "Nombre:" is followed by a text input field. Below that, the text "Apellido:" is followed by another text input field. At the bottom of the form is a "Submit" button.

Figura 5 - Ejemplo de formulario

Para el desarrollo de la aplicación, se usará concretamente HTML5. Ésta es la última revisión del lenguaje web HTML e incluye las siguientes novedades [10]:

- Incorpora etiquetas (canvas 2D y 3D, audio, vídeo) con codecs para mostrar los contenidos multimedia.
- Etiquetas para manejar grandes conjuntos de datos: Datagrid, Details, Menu y Command. Permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.
- Mejoras en los formularios. Nuevos tipos de datos (eMail, url, datetime ...) y facilidades para validar el contenido sin Javascript.
- Visores: MathML (fórmulas matemáticas) y SVG (gráficos vectoriales). En general se deja abierto a poder interpretar otros lenguajes XML.
- Drag & Drop. Nueva funcionalidad para arrastrar objetos como imágenes.
- Añade etiquetas para manejar la Web Semántica (Web 3.0): header, footer, article, nav, time (fecha del contenido), link rel="" (tipo de contenido que se enlaza).
- Estas etiquetas permiten describir cuál es el significado del contenido. Por ejemplo, su importancia, su finalidad y las relaciones que existen. No tienen especial impacto en la visualización, se orientan a buscadores.
- Los buscadores podrán indexar e interpretar esta meta información para no buscar simplemente apariciones de palabras en el texto de la página.
- Permite incorporar a las páginas ficheros RDF / OWL (con meta información) para describir relaciones entre los términos utilizados.
- Además, ofrece versatilidad en el manejo y animación de objetos simples, imágenes etc.

CSS



Figura 6 – Logo CSS3

CSS (*Cascading Style Sheets*) es un lenguaje usado para definir el estilo de los documentos HTML, es decir, mientras que con HTML definimos el contenido, con CSS definimos como va a mostrarse ese contenido. La separación entre contenido y estilo, permite cambiar por completo el aspecto de una página web, simplemente aplicando distintas hojas de estilo.

Para el desarrollo de la aplicación, se usará concretamente CSS3. Ésta es la última revisión del lenguaje. Las funcionalidades que ofrece el lenguaje CSS son [11]:

- Propiedades de la fuente, como tipo, tamaño, énfasis...
- Color de texto, fondos, bordes u otros elementos.
- Atributos del texto, como espaciado entre palabras, letras, líneas, etcétera.
- Alineación de textos, imágenes, tablas u otros.
- Propiedades de caja, como margen, borde, relleno o espaciado.
- Propiedades de identificación y presentación de listas.

Javascript



Figura 7 - Logo Javascript

Javascript es un lenguaje interpretado lanzado en 1997, usado principalmente en el lado del cliente. Este nos permite el desarrollo de páginas dinámicas mediante el procesado de los inputs del usuario y las respuestas del servidor, y la manipulación del DOM (Document Object Model) de la página. Las características de este lenguaje son [12]:

- Imperativo y estructurado: JavaScript es compatible con gran parte de la estructura de programación de C.
- Tipado dinámico: Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable. Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una cadena.
- Objetual: JavaScript está formado casi en su totalidad por objetos
- Funcional: Las funciones son tratadas como objetos, como tal, poseen propiedades y métodos
- Evaluación en tiempo de ejecución: JavaScript incluye la función “eval” que permite evaluar expresiones como expresadas como cadenas en tiempo de ejecución.
- Prototipos: JavaScript usa prototipos en vez de clases para el uso de herencia.
- JavaScript también es compatible con expresiones regulares de una manera similar a Perl.

Junto con javascript se empleará AJAX (Asynchronous JavaScript And XML). AJAX permite la comunicación asíncrona con el servidor, de forma que el usuario puede seguir interactuando con la página mientras las peticiones son resueltas. Con esta técnica podemos realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en la aplicación. También libera de algo de carga al servidor, ya que no se ve obligado a renderizar y enviar la página html completa con cada petición.



Figura 8 - Logo JQuery

Para el proyecto también se hará uso de la librería javascript JQuery y JQuery UI. Ésta permite la manipulación del documento HTML, manejo de eventos, animaciones y Ajax de una forma mucho más simple a través de una API fácil de usar y compatible con la mayoría de navegadores. JQuery UI por su parte, ofrece un conjunto funcionalidades para las interacciones con la interfaz de usuario, efectos, widgets y temas.

Bootstrap



Figura 9 - Logo Bootstrap

Bootstrap es un framework de código abierto, cuyo objetivo es el facilitar el desarrollo de webs responsivas y compatibles con dispositivos móviles. Es compatible con la mayoría de exploradores web y soporta HTML5 y CSS3. Básicamente consiste en una serie de hojas de estilo que implementan las componentes de la herramienta. Empleando los estilos definidos por bootstrap nos aseguramos por ejemplo de que nuestra página web se adaptará de forma dinámica a distintas resoluciones de pantalla.

Un ejemplo básico y fundamental de bootstrap es su sistema de cuadrícula. Consta de las siguientes clases:

- xs: (para teléfonos)
- sm (para tablets)
- md (para portátiles)
- lg (para equipos de sobremesa)

Por ejemplo, si quisiéramos generar una página web cuya estructura básica fuera una fila con tres columnas del mismo tamaño, para ordenadores portátiles tendríamos el siguiente código:

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4">.col-md-4</div>
</div>
```

Con este mismo código ejecutado en un móvil, las columnas se apilarían automáticamente. Como vemos este sistema libera de trabajo al programador, no teniendo que escribir código según el dispositivo.

4.1.2 Back-end

Para el servidor de la aplicación se ha elegido un sistema WAMP. WAMP es un acrónimo que hace referencia a las herramientas:

- Windows, como sistema operativo.
- Apache, como servidor web.
- MySQL, como gestor de bases de datos.
- PHP, como lenguajes de programación.

En concreto se ha elegido WampServer, el cual engloba y se encarga de instalar todos los programas necesarios para montar el servidor web y administrarlo.



WampServer

Figura 10 - Logo WampServer

Veamos las características y función de cada uno de los programas que conformarán el lado del servidor:

Apache



Figura 11 - Logo Apache

Apache [13] es el programa que funciona como servidor web. Apache es un proyecto de código abierto y uso gratuito, multiplataforma (hay versiones para todos los sistemas operativos más importantes), muy robusto y que destaca por su seguridad y rendimiento. Es el servidor más utilizado a nivel mundial, muy por delante de otras soluciones. Esto hace que esté ampliamente probada su seguridad y eficiencia, por lo que encaja para el desarrollo de la aplicación.

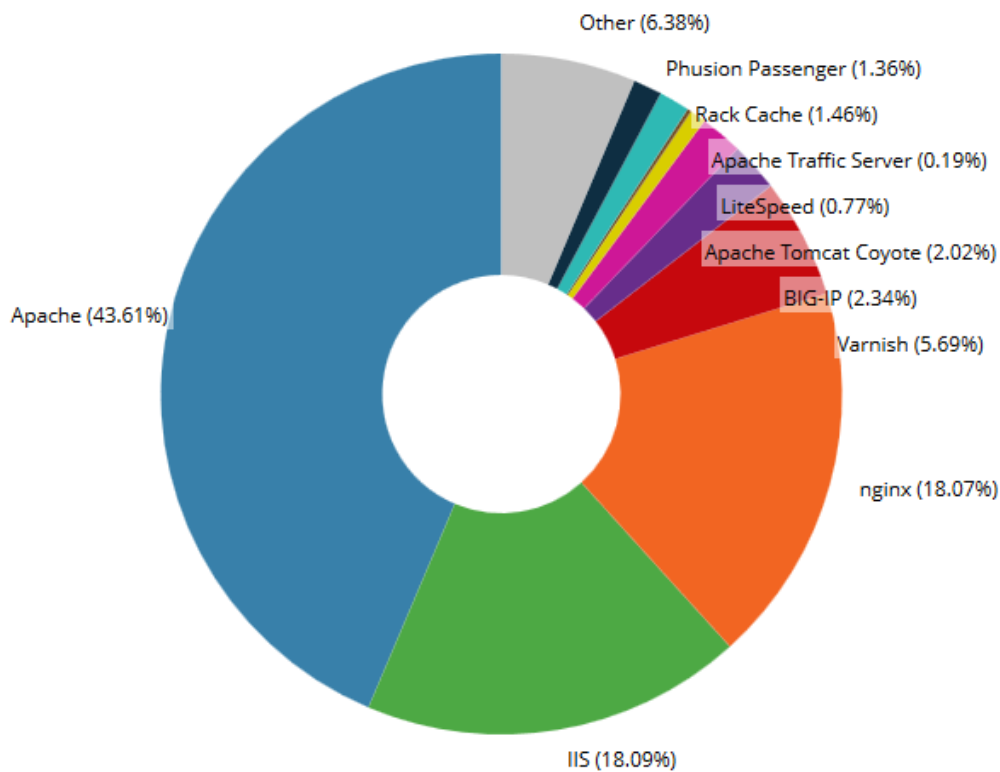


Figura 12 - Utilización de servidores

La misión del servidor web es de vital importancia, ya que es el encargado de aceptar las peticiones de páginas provenientes de los usuarios que desean acceder a nuestro sitio web y gestionar su entrega o denegación, de acuerdo a las políticas de seguridad establecidas. En general las funciones que tiene el servidor web son:

- Atender de manera eficiente, ya que puede recibir un gran número de peticiones HTTP y de forma concurrente.
- Ofrece restricciones de acceso a los ficheros que no se quieran exponer.
- Gestión de autenticaciones de usuarios o filtrado de peticiones según el origen de éstas.
- Manejar los errores por páginas no encontradas, informando al visitante y/o redirigiendo a páginas predeterminadas.
- Gestión de la información a transmitir en función de su formato e informar adecuadamente al navegador que está solicitando dicho recurso.
- Gestión de logs, como pueden ser los de acceso o los de errores.

MySQL



Figura 13 - Logo MySQL

MySQL [14] es el sistema de gestión de bases de datos que se empleará para la aplicación. Este sistema emplea el modelo relacional, que actualmente es el más utilizado y extendido. Las principales características que ofrece MySQL son:

- Compatibilidad entre plataformas.
- Los procedimientos almacenados.
- Disparadores.
- Las vistas actualizables.
- DDL en línea cuando se utiliza el motor de almacenamiento InnoDB.
- esquema de información.
- Esquema de rendimiento que recopila y agrega información sobre la ejecución del servidor y el rendimiento de las consultas con fines de seguimiento.
- Un conjunto de modos SQL para controlar el comportamiento de tiempo de ejecución, incluyendo un modo estricto para adherirse mejor a los estándares SQL.
- Procesamiento de transacciones de apoyo distribuidas(DTP).
- Transacciones con puntos de rescate cuando se utiliza el motor de almacenamiento InnoDB.
- Soporte SSL.
- Almacenamiento en caché.
- SELECTs anidados.
- El soporte integrado de replicación (es decir, maestro-maestro de replicación y maestro-esclavo de replicación) con un maestro por esclavo, muchos esclavos por maestro.
- La indización y buscado de texto.
- Biblioteca de base de datos integrada.

- El soporte Unicode.
- Agrupación compartida a través de MySQL Cluster.
- Motores nativos de almacenamiento InnoDB, MyISAM, Merge, heap, Federated, Archive, CSV, Blackhole, NDB Cluster.

PHP



Figura 14 - Logo php

PHP [15] es un lenguaje de programación empleado en el lado del servidor, principalmente diseñado para el desarrollo de aplicaciones web, ya que permite incorporar código HTML. Es compatible con la mayoría de servidores web y su sintaxis es muy parecida a lenguajes estructurados como C. Las principales características de este lenguaje son:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos.
- Es software libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables, aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones.

- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

4.2 Laravel framework

Dentro de los frameworks de desarrollo en PHP, Laravel [16] es el más popular. También forma parte del back-end, pero ya que es el framework con el cual se desarrollará la aplicación, veremos más en detalle algunas de sus características principales.



Figura 15 - Logo Laravel

4.2.1 Características generales

Veamos algunas de las características de este framework:

- Laravel tiene una sintaxis muy expresiva y clara.
- Aporta un conjunto de funcionalidades ya implementadas que permiten el ahorrar bastante tiempo de trabajo y confiar en que están bien implementadas.
- Fluent proporciona un sistema de construcción de consultas para la base de datos. Su sintaxis permite escribirlas de forma sencilla y natural. Además, utiliza vinculación de parámetros PDO (PHP Data Objects), lo que protege la aplicación contra ataques de inyección SQL. Esto ahorra el tener que limpiar las sentencias SQL.
- Ofrece el ORM (Object-Relational mapping) Eloquent. De esta forma cada tabla de la base de datos tiene un modelo correspondiente y es a través de éste la forma de interactuar con la base datos.
- Consola Artisan. Con esta herramienta podemos ejecutar mediante consola funciones útiles como son crear y correr migraciones, creación de modelos, controladores, vistas, etc.

4.2.2 Patrón de diseño software

Laravel utiliza el patrón de arquitectura software Modelo-Vista-Controlador. Con este patrón se pretende separar por un lado la lógica de la aplicación, del contenido y cómo se accede e interactúa con éste. Veamos sus tres componentes:

- **Modelo:** Los modelos representan a los objetos que tendremos en nuestra aplicación, normalmente ligados a nuestra base de datos. En ellos se definen las reglas y forma de interactuar con ellos, y permiten consultarlos, modificarlos, eliminarlos, etc.
- **Vista:** Es la parte que ven y con la que interactúan los usuarios. Es el resultado del renderizado de los datos, procedentes del servidor, por parte del navegador web.
- **Controlador:** El controlador sirve como unión entre las vistas y los modelos. Éste contiene la lógica necesaria para intercomunicar ambos de la forma deseada y llevar a cabo las acciones correspondientes en cada momento.

El flujo que se sigue con este patrón es:

1. Un usuario interactúa con la interfaz de usuario, que sería la vista.
2. El sistema de rutas redirecciona al controlador. La interacción ha generado eventos, los cuales son recogidos por el controlador.
3. En función del evento que ha recibido interaccionará con los datos correspondientes a través de los modelos. Éstos se encargarán de consultar la base de datos.
4. Por último, se encargará de enviar o insertar en la vista los nuevos datos, y ésta los renderizará para poder hacerlos visibles al usuario.

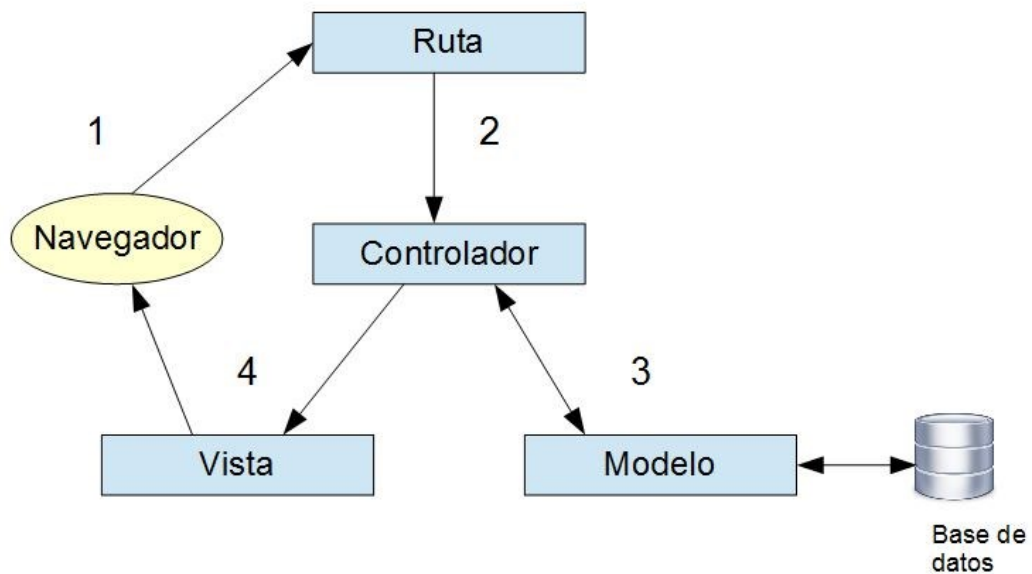


Figura 16 - Patrón MVC

Al separar las distintas partes de la aplicación siguiendo este patrón, se facilita el desarrollo, comprensión y mantenibilidad del código generado.

4.2.3 Routing

Laravel cuenta con un sistema para manejar las rutas URL de la aplicación, para ello hace uso de un helper incluido con el framework. Este sistema de rutas permite mapear las peticiones HTML de tipo GET, POST, PUT, PATCH y DELETE, de forma que puede asignarse a una ruta una acción determinada, como pueden ser devolver una vista, llamar a una función de un controlador, redirigir a otra ruta, etc.

Veamos un simple ejemplo donde simplemente nos devuelve el texto “Buenos días”:

```
Route::get('saludo', function () {

    return 'Buenos días';

});
```

Por ejemplo, el código que se encuentra a continuación, al acceder a la ruta parcial "/" se ejecuta la función definida en el controlador HomeController.

```
Route::get('/', 'HomeController@root');
```

Otro ejemplo correspondiente a la ruta para el login:

```
$this->get('login', 'Auth\AuthController@showLoginForm');
```

Aparte de los sencillos ejemplos que se han mostrado, el sistema de rutas es realmente potente. Permite el uso de parámetros en las rutas, expresiones regulares, crear agrupaciones de rutas que compartan middleware o definir funciones in-line en el mapeado de las mismas.

4.2.4 Plantillas Blade

Las vistas en Laravel se pueden escribir usando Blade. Blade es un motor de plantillas simple pero potente, proporcionado con Laravel. A diferencia de otros motores de plantillas PHP populares, Blade no restringe el uso de código PHP plano en las vistas. Todas las vistas Blade se compilan en código PHP y son guardados en caché hasta que sean modificadas, por lo que apenas añade procesado extra a la aplicación. Las vista Blade utilizan la extensión de archivo ".blade.php" y se almacenan en el directorio Resources/views.

Dos de las principales ventajas de usar Blade, son que permite la herencia de plantillas y de secciones, lo que conlleva un mejor modularidad de las vistas. Por ejemplo, al definir una página hija, es posible utilizar la directiva @extends para especificar qué diseño de la página (layout) debe heredar. Para definir el contenido susceptible a ser inyectable en otras vistas utilizamos la directiva @section. Luego para incluir estas secciones en otra vista usamos la directiva @yield.

Veamos un ejemplo para una mayor claridad. Primero creamos nuestro layout:

```
<!DOCTYPE html>

<head></head>

<body id="app-layout">

  <h1>Mi Layout</h1>

  @yield('content')

</body>

</html>
```

A continuación, definimos una sección con código HTML que deseamos insertar dentro del layout 'app':

```
@extends('layouts.app')

@section('content')

<div>

  <p>Texto de ejemplo</p>

</div>

@endsection
```

Como vemos en el ejemplo, hemos definido una sección llamada 'content' usando @section('content'). Ésta se introducirá en el layout donde aparece la sentencia @yield('content').

4.2.5 Middleware

Laravel también provee un middleware HTTP, el cual proporciona mecanismos para el filtrado de las solicitudes HTTP entrantes. Entre otros, incluye un middleware para la verificación del autenticado del usuario en la aplicación. Si el usuario no está autenticado, el middleware se redirigirá al usuario a la pantalla de inicio de sesión. Sin embargo, si el usuario se autentica correctamente, el middleware permitirá proceder adelante en la aplicación.

Laravel hace que la implementación de la autenticación sea muy simple. El archivo de configuración de la autenticación se encuentra en `config/auth.php`, y contiene varias opciones bien documentadas para configurar el comportamiento de los servicios de autenticación.

Algunas de las facilidades que proporciona para la autenticación son los guards y los providers. Los guards definen cómo los usuarios se autentican para cada solicitud. Por ejemplo, el `session guard`, el cual mantiene el estado usando sesiones y cookies, y también utiliza un token enviado en cada petición para mantener el autenticado. Los providers definen como se recuperan los datos de los usuarios del almacenamiento persistente (base de datos).

Dentro de los controladores para la autenticación, el `AuthController` se encarga de manejar los nuevos registros y de la autenticación de los usuarios, mientras que el `PasswordController` contiene la lógica para ayudar a los usuarios existentes restablecer sus contraseñas olvidadas.

4.2.6 Migraciones

Las migraciones funcionan como el control de versiones de la base de datos, lo que permite a un equipo modificar y compartir el esquema de la base de datos de la aplicación de forma sencilla. Las migraciones suelen estar emparejadas con el constructor del esquema de Laravel para construir fácilmente el esquema de base de datos de la aplicación. Una clase de tipo migración contiene dos métodos: `up` y `down`. El método `up` se utiliza para agregar nuevas tablas, columnas o índices para su base de datos, mientras que el método `down` debe revertir las operaciones llevadas a cabo por el método `up`.

4.2.7 Protección CSRF

Laravel facilita la protección ante ataques CSRF (cross-site request forgery). Este ataque consiste en un *exploit* malicioso de un sitio web, en el cual comandos no autorizados son transmitidos por un usuario autenticado en el cual la aplicación web confía.

Para solucionarlo, se genera automáticamente un "token" CSRF para cada sesión de usuario activo administrado por la aplicación. Este token se utiliza para verificar que el usuario autenticado es el que realmente hace las peticiones a la aplicación.

Para hacer efectiva esta protección, en cualquier lugar que se defina un formulario HTML en la aplicación, se debe incluir un campo token CSRF oculto, para que el middleware de protección CSRF sea capaz de validar la solicitud. Para generar un token oculto que contenga el token CSRF, sería algo así:

```
<form action="/foo/bar" method="POST">
    <input type="hidden" name="_method" value="PUT">
    <input type="hidden" name="_token" value="{{ csrf_token() }}">
</form>
```

No es necesario para verificar manualmente el token CSRF en peticiones POST, PUT, DELETE, etc. El middleware `VerifyCsrfToken`, verificará automáticamente que el token se encuentra incluido en la de solicitud y coincide con la señal almacenada en la sesión.

4.3 Base de datos

Para el desarrollo de la base de datos se ha empleado un modelo relacional junto con la herramienta MySQL. Para cumplir con los requisitos del sistema y ofrecer las funcionalidades descritas, la base de datos constará de las tablas:

- **Users:** En esta tabla se almacenan los datos de los usuarios registrados en la aplicación, como pueden ser su nombre, email o contraseña de acceso. También se almacena su rol en la aplicación que puede ser estudiante o administrador.
- **Solutions:** En esta tabla se almacenan las respuestas correctas a los ejercicios de cada unidad, así como la solución de los mismos con la que se informará al usuario.
- **Progress:** Esta tabla es necesaria para llevar el control del progreso de cada alumno en el contenido docente. En ella se almacena para cada alumno el último objetivo superado y su unidad correspondiente.
- **Statistics:** En esta tabla se almacenarán las estadísticas de los alumnos. En concreto se guardarán los fallos cometidos por objetivo y unidad, así como los tiempos empleados para superar cada objetivo.
- **Bugs:** En esta tabla se almacenarán los bugs reportados por los alumnos. Se identificarán por medio de la unidad y objetivo donde se haya reportado, y contará con una descripción dada por el usuario.
- **Migrations:** En esta tabla se almacenarán las migraciones de base de datos ejecutadas. Sirve como un control de versiones de la base de datos, permite roll-backs a estados anteriores.
- **Password_resets:** La función de esta tabla es servir en el proceso de restablecer la contraseña de un usuario que la haya olvidado y llevar registro de ello.

El diagrama de la base de datos quedaría:

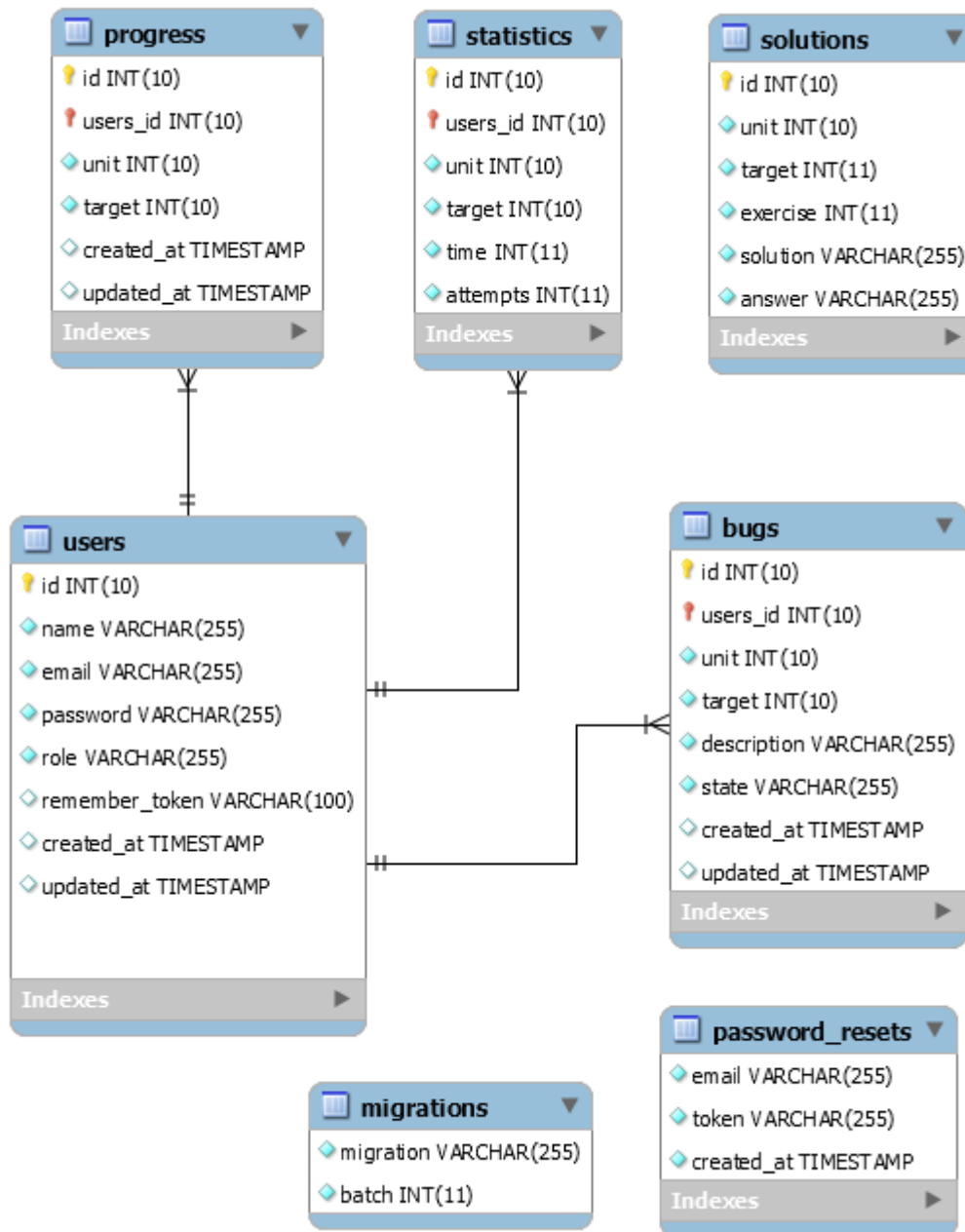


Figura 17 - Esquema de la base de datos

4.4 Contenido docente

El contenido docente estará dividido en unidades (ver índice de unidades en el Anexo A). A su vez, las unidades se encontrarán conformadas por varios objetivos a modo de pequeñas píldoras de conocimiento. Los objetivos consistirán en una breve lección que el alumno deberá leer. Dicha lección, normalmente, irá seguida de un pequeño ejercicio referente a la misma.

La idea general es ir introduciendo y familiarizando al alumno poco a poco. Al principio será normal que se encuentre con ejercicios que consistan, básicamente, en observar que salida produce un código ya dado, o simplemente vea el estilo con el que está escrito. La complejidad de los ejercicios irá aumentando gradualmente y el alumno habrá ido superando pequeños ejercicios a medida que avanza. De este modo, al ir avanzando entre objetivos y unidades, se pretende que el estudiante no pierda la perspectiva de todo lo visto anteriormente, ya, que el mismo, habrá ido completando objetivos que le han llevado hasta ese punto. De esta forma, se busca que el aprendizaje sea más interactivo y el alumno sea una parte activa del mismo.

Por otra parte, el objetivo de la aplicación no es el calificar ni evaluar al alumno en función de los resultados obtenidos en ella. Sabiendo esto el alumno, se elimina la ansiedad que puede crearle el aprobar, a la vez que se evita el buscar la solución fácil de copiar al compañero para completar los objetivos. También permitirá al alumno llevar su propio ritmo de aprendizaje, y no uno impuesto externamente. De esta forma, se busca que sea usada con el único propósito de querer aprender.

Este enfoque de aprendizaje contrasta con el típico y más extendido, en el que al alumno sencillamente se le da un contenido bastante amplio, que debe estudiar de golpe y luego ponerlo a prueba. Este enfoque tradicional suele causar dejadez por parte de los estudiantes, y cuando quieren ponerse al día el esfuerzo que deben realizar es enorme.

5 Implementación

En este capítulo, se verá el resultado final del desarrollo de la aplicación, siguiendo el diseño expuesto en el capítulo anterior.

5.1 Estructura de la aplicación

La aplicación en su totalidad se encuentra alojada en el servidor web, y éste enviará al cliente aquellos elementos que le sean necesarios para la ejecución de la misma en su navegador. A continuación, se verá cómo está estructurada. En la raíz del proyecto tendremos los siguientes directorios:

- **App:** Este directorio contiene el código del núcleo de la aplicación. Contiene las clases que sirven de modelos de las tablas de la base de datos. Dentro también encontramos los directorios:
 - Console: Contiene los comandos para Artisan, que es la consola incluida con Laravel.
 - Http: Contiene los controladores del back-end, middleware y peticiones.
 - Events: Contiene las clases para la definición de eventos.
 - Exceptions: Contiene las implementaciones de las clases encargadas de capturar las excepciones.
 - Jobs: Contiene las tareas a realizar por la aplicación
 - Listeners: Contiene las implementaciones de las clases encargadas de capturar los eventos.
 - Policies: Contiene las políticas de autenticación de la aplicación. Aquí se definen las acciones que puede realizar cada tipo de usuario.

- **Bootstrap:** Este directorio contiene archivos encargados de arrancar y configurar el framework, así como un directorio de caché que contiene algunos archivos para la optimización del rendimiento de arranque.
- **Config:** Este directorio contiene todos los archivos de configuración de la aplicación.
- **Database:** Este directorio contiene todas las migraciones e inicializadores para la base de datos de la aplicación.
- **Public:** Este directorio contiene los controladores del front-end y sus assets (imágenes, JavaScript, CSS, etc.).
- **Resources:** Este directorio contiene las vistas y sus layouts, es decir los ficheros HTML.
- **Storage:** Este directorio contiene compiladas las plantillas Blade, ficheros de sesiones, ficheros para la caché y otros archivos generados por el framework. Aquí también se almacenan ficheros como pueden ser los correspondientes a los contenidos del módulo docente, o los ficheros temporales de ejecución y compilación de C. Dentro de este directorio se crean carpetas para los ficheros y ejecutables generados por cada alumno registrado en la aplicación.
- **Tests:** Este directorio contiene pruebas automatizadas, test unitarios, etc.
- **Vendor:** Este directorio contiene las dependencias de Composer. Composer es un manejador de dependencias, el cual se encarga de instalar los paquetes y librerías necesarias.

5.2 Módulo docente

En esta sección se va a detallar la información referente al módulo docente, es decir, la parte con la que interactuarán los alumnos. Por lo tanto, se mostrará y explicará la interfaz de usuario desarrollada, así como las soluciones de implementación llevadas a cabo.

5.2.1 Compilación, ejecución e interacción con el código c

Una de las características centrales del proyecto es la compilación y ejecución de código C desde el navegador web, liberando a los estudiantes de tener que instalar y configurar un entorno de programación específicamente para ello. Esto les animará y permitirá poder empezar a programar desde el primer día, y les ahorrará tiempo y posibles problemas con los entornos de programación.

En lo referente al editor de texto y la interfaz con la que interactuarán los usuarios a la hora de programar, se ha buscado la máxima simplicidad de uso a la vez que una experiencia lo más cercana a la realidad posible. A continuación, se muestra una imagen de la interfaz para la programación:

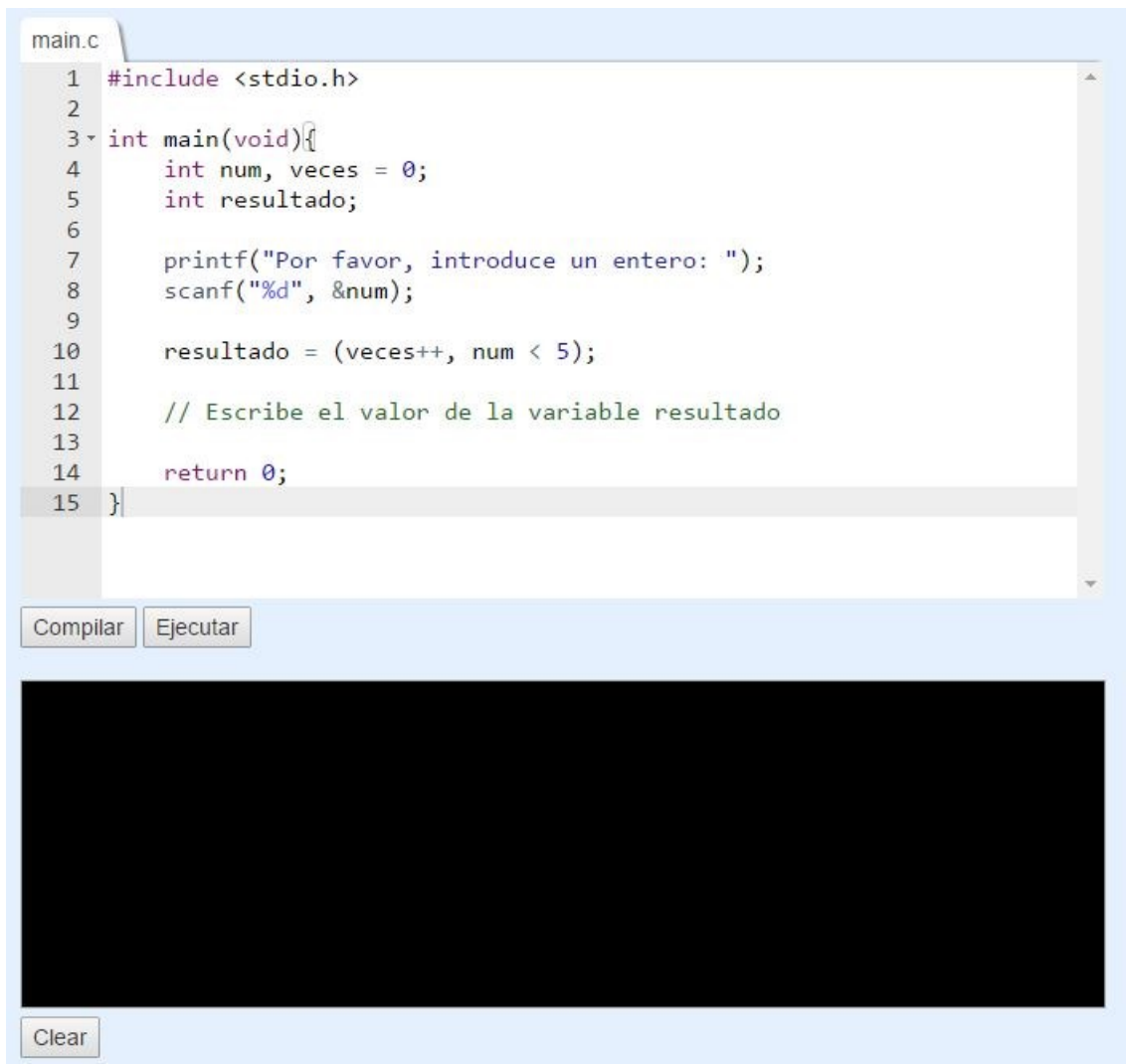


Figura 18 – Interfaz para la programación

Como puede observarse en la imagen, en primer lugar, nos encontramos con el editor de texto. Se ha tratado de ofrecer a los alumnos un editor de texto funcional a la vez que cómodo y potente. Este editor cuenta con múltiples ayudas y funcionalidad que hace su uso más atractivo y sencillo a la hora de escribir código. Algunas de sus características son [17]:

- Indentación automática.
- Manejo de documentos de gran tamaño
- Combinaciones de teclas personalizables
- Búsqueda y reemplazo con expresiones regulares
- Resaltado de los paréntesis emparejados
- Mostrar caracteres ocultos
- Permite arrastrar y soltar texto
- Justificación de línea
- Plegado de código
- Múltiples cursores
- Comprobador de sintaxis

A continuación, bajo el editor, nos encontramos con dos botones, compilar y ejecutar. Estos se podrían haber fusionado en uno único, pero de esta forma se pretende que los alumnos sean conscientes y se familiaricen con algo tan fundamental y diferenciado como es el proceso de compilación y ejecución de programas.

```
main.c | solution
1 #include <stdio.h>
2
3 int main(void){
4
5     int altura, aux;
6     double peso, imc;
7
8     printf ("Por favor, introduce tu altura en cm: ");
9     scanf ("%d", &altura);
10
11     printf ("Por favor, introduce tu peso en kg: ");
12     scanf ("%lf", &peso);
13
14     imc = 10000 * peso / (altura * altura);
15
16     if (imc < 18.5)
```

Compilar Ejecutar

```
gcc -Wall -o main main.c
./program.c: In function 'main':
./program.c:5:14: warning: unused variable 'aux' [-Wunused-variable]
int altura, aux;
            ^
```

Figura 19 - Ejemplo de compilación

Por último, nos encontramos con un editor de fondo negro. Este editor simula la típica terminal del sistema y su objetivo es tanto informar del proceso de compilación, con sus posibles avisos y errores en caso de haberlos, como de mostrar el resultado de la ejecución como se vería en cualquier terminal típica. Esta terminal también permite la inserción de inputs del usuario para su programa como en cualquier otra terminal, escribiéndolo y pulsando enter.

```
main.c
7   printf ("Por favor, introduce tu altura en cm: ");
8   scanf ("%d", &altura);
9
10  printf ("Por favor, introduce tu peso en kg: ");
11  scanf ("%lf", &peso);
12
13  imc = 10000 * peso / (altura * altura);
14
15  if (imc < 18.5)
16      printf ("Tu peso está por debajo de lo normal\n");
17  else
18      printf ("Tu peso no está por debajo de lo normal\n");
19
20  return 0;
21 }
```

Compilar Ejecutar

```
Por favor, introduce tu altura en cm: 186
Por favor, introduce tu peso en kg: 67
Tu peso no está por debajo de lo normal
|
```

Figura 20 - Ejemplo de ejecución

Proceso de compilado

C es un lenguaje compilado, por lo tanto, el primer paso para poder ejecutar el programa será la compilación del mismo. El código escrito por el usuario es enviado al servidor, el cual lleva a cabo los siguientes pasos una vez recibido el código:

1. Análisis y búsqueda de palabras o estructuras de código mediante expresiones regulares para:
 - Borrado por motivos de seguridad, rendimiento e integridad del servidor. Por ejemplo llamadas al sistema (`system(...)`), bucles infinitos (`while(1)...`)

- Inyección de código de control. Esto es necesario para poder llevar a cabo el control de flujo entre los inputs del usuario y la respuesta del servidor a estos inputs. Se lleva a cabo con funciones del tipo scanf(...), etc.
2. Creación del fichero .c en la carpeta correspondiente al usuario que envió el código.
 3. Llamada al sistema para ejecutar el compilador GCC. El servidor realiza una llamada al sistema para ejecutar el compilador de c (GCC) y guarda la respuesta de esta llamada
 4. Responde al usuario con el valor devuelto por la llamada al compilador.

Proceso de ejecución

Una vez compilado el código, se ha creado un ejecutable en la carpeta correspondiente del usuario logueado. El proceso de ejecución comprende:

1. Acceso a la ruta correspondiente al usuario.
2. Llamada al sistema para ejecutar el .exe generado al compilar, redireccionando su salida a un fichero.
3. Lectura del fichero con la salida del programa y envío de respuesta.

5.2.2 Contenido docente

Otro de los puntos centrales del proyecto es el referente al contenido docente, su forma de mostrarlo al alumno, control de su avance, ejercicios propuestos y corrección de los mismos. Veámoslo en conjunto:

Índice C-Learning Andrés ▾

1. Buenas tardes

El programa de esta unidad te puede venir bien si estás pensando viajar al extranjero.
El programa convierte dólares, yenes y libras a euros.
Como en anteriores unidades, empezamos por ejecutarlo para luego ir construyéndolo paso a paso.

Ejercicio: Ejecuta el programa para transformar 150 libras a euros.
¿Qué cambio proporciona el programa?

a. 126.53 euros
 b. 173.41 euros
 c. 178.31 euros

```
main.c
1 #include <stdio.h>
2
3 int main(void) {
4     const double DOLAR = 1.343;
5     const double YEN = 102.961;
6     const double LIBRA = 0.865;
7     double cantidad;
8     int moneda;
9
10    printf("CONVERSION DE MONEDA EN EUROS\n");
11    printf(" 1. Dólar\n");
12    printf(" 2. Yen japonés\n");
13    printf(" 3. Libra esterlina\n");
14
15    printf("Introduce una cantidad: ");
16    scanf("%lf", &cantidad);
```

« 1 2 3 4 5 6 7 8 »

Escuela Politécnica Superior

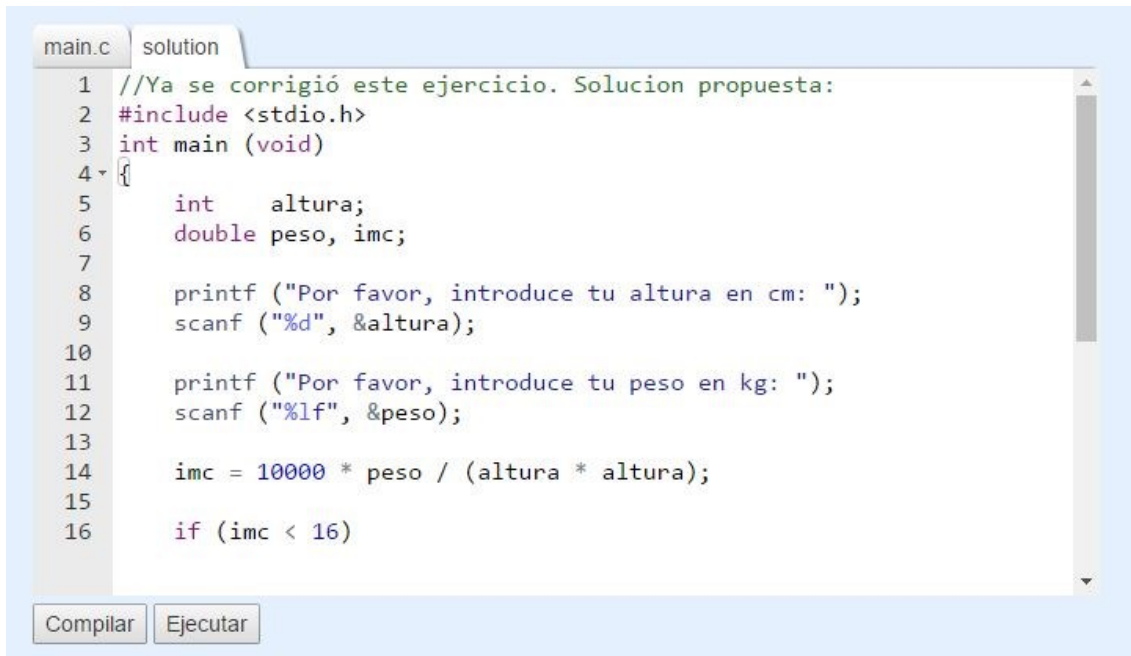
Figura 21 - Pantalla completa del módulo docente

Como puede observarse, nos encontramos a la derecha con la parte de codificación y ejecución del código que ya analizamos anteriormente, y a la izquierda con la zona donde se muestra el material didáctico correspondiente a cada unidad.

La estructura general de cada lección consta inicialmente de una explicación del tema a tratar, seguido de un ejercicio para afianzar lo aprendido. Los ejercicios pueden ser del tipo:

- Test. En estos se mostrará varias opciones posibles con una única respuesta correcta.
- Modificar un programa. Se pedirá que modifiquen o completen un programa, ya cargado en el editor, para que realice la función solicitada.
- Desarrollar un programa completamente. En estos casos el alumno deberá escribir un programa que cumpla con los requisitos solicitados. Estos programas se corregirán automáticamente en el servidor con inputs de prueba y se comprobará que la salida es la esperada.

Al completar cada ejercicio se informará al alumno de su acierto o fallo, y se le informará de la solución correcta. En el caso de desarrollar programas, la solución propuesta le aparecerá en una nueva pestaña en el editor de código. De esta forma el alumno obtiene feedback continuamente, lo que le ayudará a afianzar los contenidos.



```
main.c solution
1 //Ya se corrigió este ejercicio. Solucion propuesta:
2 #include <stdio.h>
3 int main (void)
4 {
5     int altura;
6     double peso, imc;
7
8     printf ("Por favor, introduce tu altura en cm: ");
9     scanf ("%d", &altura);
10
11     printf ("Por favor, introduce tu peso en kg: ");
12     scanf ("%lf", &peso);
13
14     imc = 10000 * peso / (altura * altura);
15
16     if (imc < 16)
```

Compilar Ejecutar

Figura 22 - Ejemplo de solución a ejercicio

Debajo del contenido didáctico de la unidad podremos encontrar el botón ayuda, cuando proceda. Éste, al pulsarlo mostrará un mensaje con información o pistas que pueden ayudar al alumno a completar el ejercicio planteado. La figura 23 muestra un ejemplo:

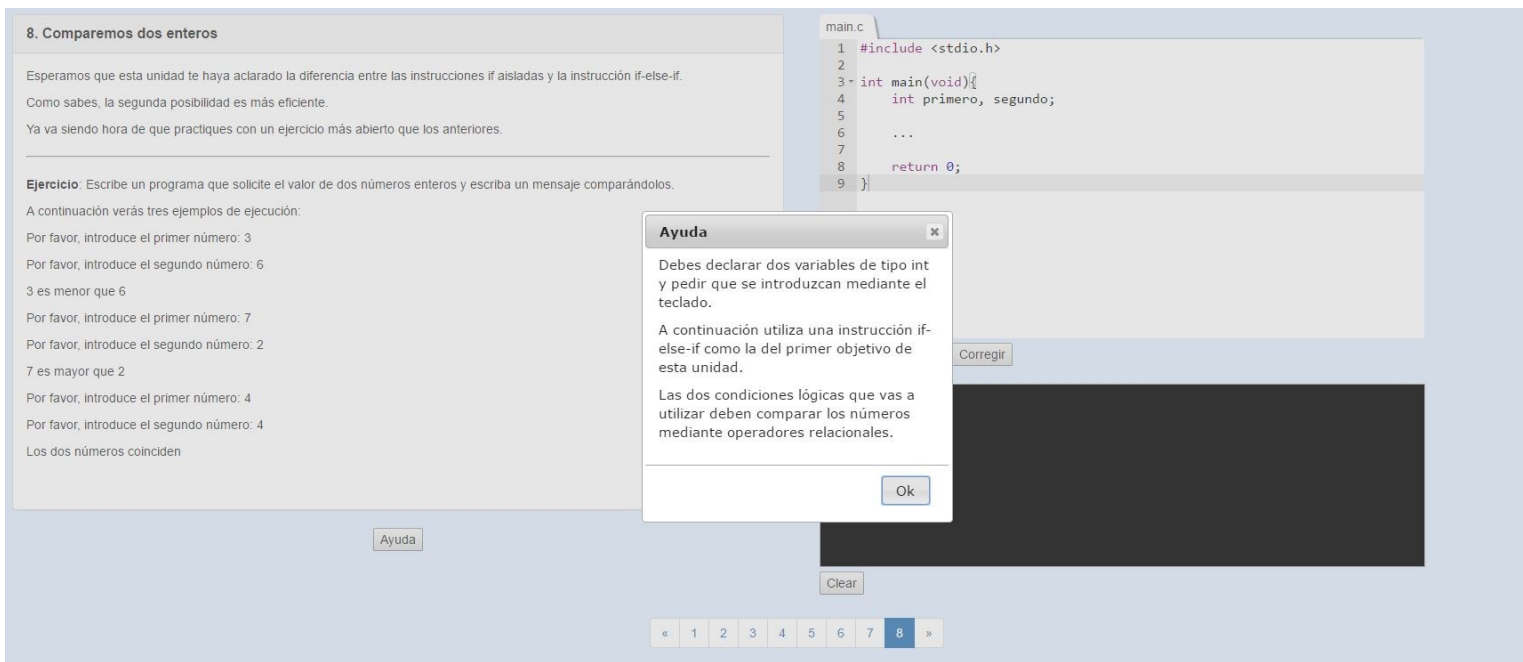


Figura 23 - Pantalla de ayuda

Por debajo de todo nos encontramos con el paginador. Con este elemento los alumnos pueden ir avanzando a lo largo de la unidad. También sirve como indicador de en qué página se encuentran actualmente, ya que esta se muestra resaltada en azul. Cada vez que se selecciona una página se actualiza tanto el contenido docente como el código asociado al mismo.

Como último elemento, los alumnos contarán con un botón que les permitirá reportar bugs que puedan encontrar en la aplicación. Este elemento es bastante importante, ya que permitirá ir corrigiendo fallos de la aplicación y haciéndola más estable.

5.3 Módulo de evaluación

Este módulo solo es visible para los profesores. El objetivo principal de éste es el presentar a los profesores la información recogida por la aplicación, a partir de los datos obtenidos de los alumnos mientras van completando el curso.

Los datos recogidos y almacenados en la base de datos, son consultados según la información que se desea conocer y procesados en el servidor para mandárselos al cliente en el formato correcto y esperado por la librería gráfica.

5.3.1 Librería gráfica

Para la presentación de los datos y las gráficas, se ha utilizado la librería Canvas.JS y su API [18]. Se trata de una librería desarrollada en javascript y HTML5 que cuenta con las siguientes características:

- Posee una sintaxis simple y sencilla, lo que permite desarrollar de forma rápida y ágil.
- Tiene un aspecto visual personalizable y bastante atractivo para el usuario.
- Permite elegir entre gran variedad de tipos de gráficos: barras, columnas, líneas, áreas, etc.
- Buen rendimiento, siendo capaz de renderizar unos 100000 puntos en aproximadamente 100 milisegundos.
- Compatibilidad con dispositivos móviles Android e iPhone.

5.3.2 Pantalla de estadísticas

Se ha tratado de presentar la información más relevante, de la forma más sencilla, intuitiva y útil posible para el profesorado. A continuación, se mostrarán y explicarán los distintos tipos de gráficos que genera la aplicación, su función y utilidad. Veamos como se muestra el panel de administración:



Figura 24 - Progreso de los alumnos

Como vemos, lo primero que nos muestra es el progreso general de los alumnos. Se trata de un gráfico de barras, el cual está ordenado de menor a mayor progreso en las unidades del curso. También refleja debajo del progreso el total de tiempo invertido en el curso. El gráfico está en este orden ya que en primer lugar aparecen los alumnos que más dificultades o mayor retraso llevan en el curso, lo que puede implicar un mayor riesgo de abandono o suspenso. Este gráfico permite una visión rápida a la situación general de los alumnos, en especial de los más rezagados, para quizás poder ponerse en contacto con ellos u ofrecerles apoyo extra.

También podemos seleccionar un alumno concreto, una unidad y ver su progreso en cada objetivo de la misma pulsando el botón "Show":

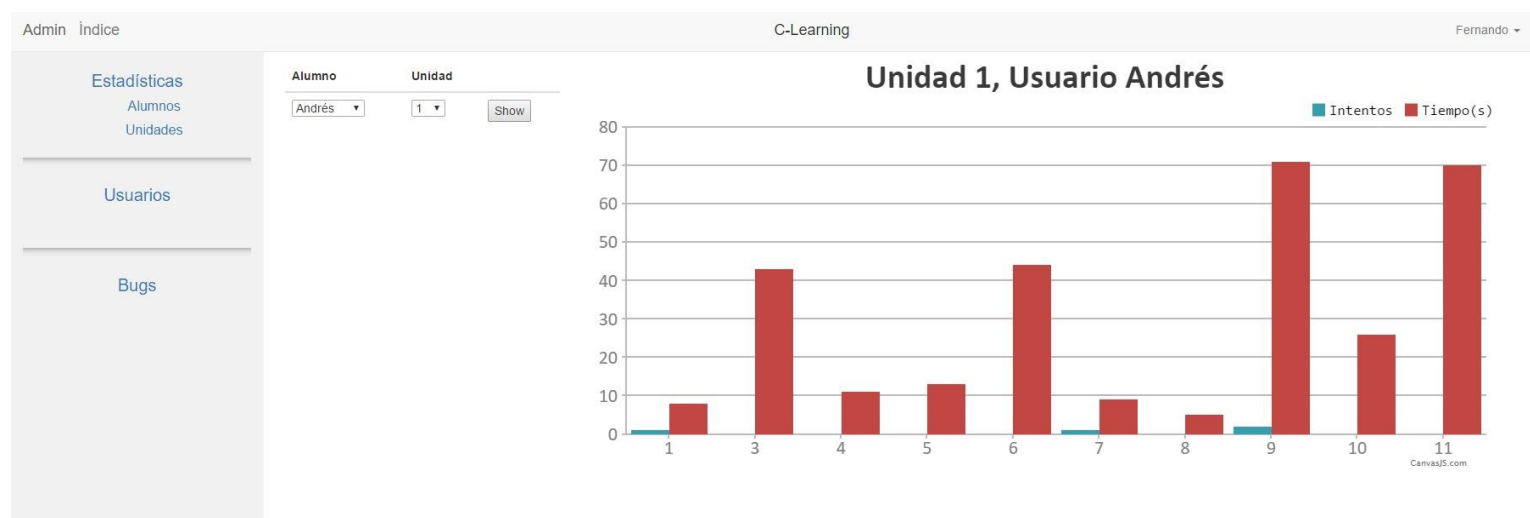


Figura 25 - Estadísticas de usuario por unidad

En cuanto a las estadísticas por unidades, se mostrará una gráfica de la siguiente forma:

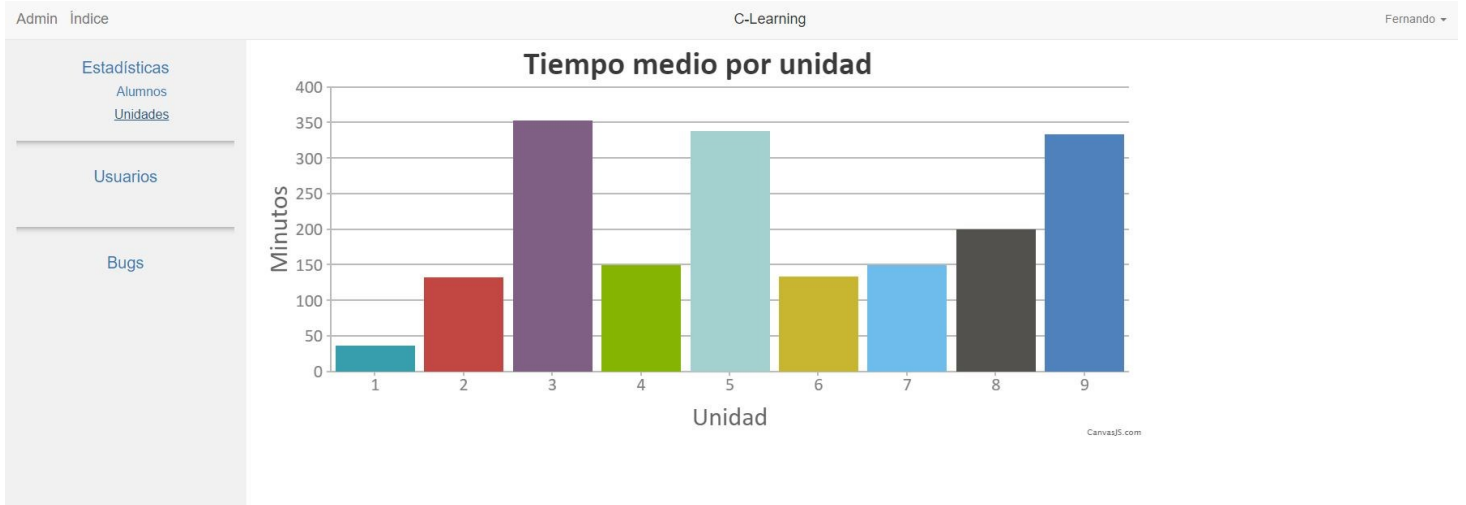


Figura 26 - Tiempo por unidad

Se trata de una gráfica de barras, en la cual se muestra el tiempo medio que han dedicado los alumnos para superar cada unidad. Estos datos pueden ser útiles para detectar dónde los alumnos se encuentran con más dificultades, o para ajustar el contenido de las unidades.

5.3.3 Pantalla de usuarios

Si pulsamos en Usuarios se muestra la siguiente página:

User Id	Nombre	Correo	Rol	Registrado	
1	Andrés	aaa@a.com	estudiante	2016-08-27 18:54:07	Editar
2	Fernando	aaa2@a.com	admin	2016-08-28 10:43:55	Editar
3	Paco	paco@cc.com	estudiante	2016-08-28 14:39:46	Editar
4	Ana	ana@cc.com	estudiante	2016-08-28 14:39:47	Editar
5	Juan	j@cc.com	estudiante	2016-08-28 14:39:49	Editar
6	Antonio	an@cc.com	estudiante	2016-08-28 14:39:51	Editar
7	María	m@cc.com	estudiante	2016-08-28 14:39:51	Editar
8	Alejandro	ax@cc.com	estudiante	2016-08-28 14:39:52	Editar
9	Mario	mu@cc.com	estudiante	2016-08-28 14:39:53	Editar

Figura 27 - Pantalla de usuarios

Nos muestra una lista con todos los usuarios registrados en nuestro sistema. En ella podemos ver los campos nombre, correo, rol en la aplicación (estudiante o administrador) y fecha en la que se registraron. También, al lado de cada usuario tenemos un botón “editar” para modificar alguno de los campos si fuera necesario.

5.3.4 Pantalla de bugs

Por último, nos encontramos con la pantalla de bugs:

Bug Id	User Id	Unidad	Objetivo	Descripción	Estado
1	1	6	1	Falla el código de ejemplo	pending <input type="button" value="Solucionado"/>
2	3	2	4	aparece mensaje de error	pending <input type="button" value="Solucionado"/>
3	5	3	6	no puedo avanzar del ejercicio 6	pending <input type="button" value="Solucionado"/>

Figura 28 - Pantalla de bugs

En ésta, encontramos en forma de lista, todos los bugs reportados por los usuarios. En concreto informa del usuario que reportó el bug, la unidad y el objetivo donde lo detectó, su propio comentario sobre el mismo y el estado en el que se encuentra. Como último elemento tenemos el botón “solucionado”, con el cual cambiaremos el estado pendiente del bug a solucionado, una vez que haya sido corregido y probado.

6 Pruebas y resultados

6.1 Validación de requisitos

Dado que se trata de un sistema complejo, es necesario someterlo a un plan de pruebas. Esto es para verificar el software (es decir, para determinar si satisface las condiciones iniciales impuestas al sistema) y para validarlo (comprobar que satisface los requisitos marcados por el usuario).

El objetivo de esta sección es llevar a cabo las pruebas sobre los requisitos funcionales del proyecto, los cuales serán analizados para evaluar su consistencia, completitud y factibilidad de forma individual. De esta forma se asegura que la aplicación cuenta con toda la funcionalidad requerida.

Requisito funcional	Prueba realizada	Resultado esperado	Resultado
RF-1	Registrar un nuevo usuario en la aplicación usando el formulario web.	Registro con los datos del usuario creados en la base de datos.	Superada
RF-2	Intentar introducir la URL de una unidad sin estar logueado.	Redirección a la página de login.	Superada
RF-3	Completar los objetivos de la primera unidad.	Comprobar en la base de datos que se ha guardado el avance.	Superada
RF-4	Avanzar dentro de una unidad.	Visualizar el contenido docente correctamente.	Superada

RF-5	Completar un ejercicio propuesto.	Informe de corrección del mismo.	Superada
RF-6	Completar uno de los tests.	Informe de corrección del mismo.	Superada
RF-7	Intentar acceder a la unidad 2 sin haber completado la 1.	Mensaje informativo para completar la anterior unidad.	Superada
RF-8	Consultar contenido de la unidad 1 una vez superada la 2.	Poder visualizar el contenido sin problema.	Superada
RF-9	Completar la unidad 1.	Registros con las estadísticas creados en la base de datos.	Superada
RF-10	Compilar un "hello world".	Mensajes del compilador.	Superada
RF-11	Ejecutar un programa escrito en C.	Salida en la consola de la aplicación.	Superada
RF-12	Escribir un programa que haga uso de la función scanf e introducir el input solicitado.	Comportamiento del programa esperado al introducir el input.	Superada
RF-13	Escribir código C en el editor web.	Visualización y manejo del código en el editor.	Superada
RF-14	Escribir y enlazar una librería junto al main.c	Compilación y ejecución correctas.	Superada
RF-15	Probar a reportar un bug.	Registro del bug creado en la base de datos.	Superada

RF-16	Comprobar gráficas generadas en el módulo de administración.	Gráfica con valores coherentes según los valores de la base de datos.	Superada
RF-17	Acceder a los usuarios a través del módulo de administración.	Usuarios mostrados coherentes con los almacenados en la base de datos.	Superada
RF-18	Acceder a los bugs a través del módulo de administración.	Bugs mostrados coherentes con los almacenados en la base de datos.	Superada

Tabla 3 - Validación de requisitos

Todas las pruebas fueron superadas, indicando que la aplicación desarrollada cumple con los requisitos previamente fijados para su completitud, por lo que cuenta con funcionalidad completa para empezar a ser usada.

6.2 Pruebas del servidor web

Ya que la aplicación va a ser utilizada por múltiples usuarios de manera concurrente, es necesario establecer y ejecutar un conjunto de pruebas, con el objetivo de conocer las capacidades y límites del servidor donde se va a encontrar alojada la aplicación web.

6.2.1 Entorno de pruebas

El equipo donde está montado el servidor web cuenta con las siguientes características técnicas:

Servidor web	
Sistema operativo	Windows 10 64bits
Procesador	Intel Core 2 Quad Q6600 @ 2,4 GHz
Memoria ram	4GBytes DDR2
Disco duro	240 GBytes
Tarjeta gráfica	Nvidia NV44 128 MBytes

Tabla 4 - Características del servidor web

Para realizar las pruebas se hará uso de la aplicación Apache JMeter [19]. JMeter permite probar el rendimiento tanto de recursos estáticos y dinámicos como son servicios web SOAP/REST, lenguajes como PHP, Java, ASP.NET, etc., objetos Java, bases de datos, servidores FTP, etc. Se puede utilizar para simular pruebas de carga en un servidor, grupo de servidores, o redes para analizar el rendimiento bajo diferentes tipos de carga.



Figura 29 - Logo Apache JMeter

Las pruebas de stress se realizarán simulando peticiones HTTP al servidor. Se simularán peticiones, de distintos números de usuarios concurrentes, en un intervalo de tiempo de 5 segundos. Como tope de usuarios se simulará con 500. Este número de usuarios es más que suficiente para comprobar su capacidad, ya que es superior al número de usuarios finales que se espera utilicen la aplicación.

6.2.2 Ejecución de las pruebas y análisis de resultados

A continuación, se mostrarán los resultados obtenidos al ejecutar las pruebas de stress contra el servidor web. Para todas las pruebas se ha configurado el periodo de subida en 5 segundos, es decir, el periodo durante el cual el servidor va a verse sometido a múltiples peticiones. El periodo de tiempo es tan pequeño para asegurar que se somete al servidor a una situación de estrés elevado, ya que de ser bastante mayor podría ir resolviendo las peticiones antes de que entrarán nuevas.

Para una primera prueba se han simulado 10 usuarios, lo cual no debería suponer un problema para el servidor. Veamos los resultados junto con la interfaz de JMeter:

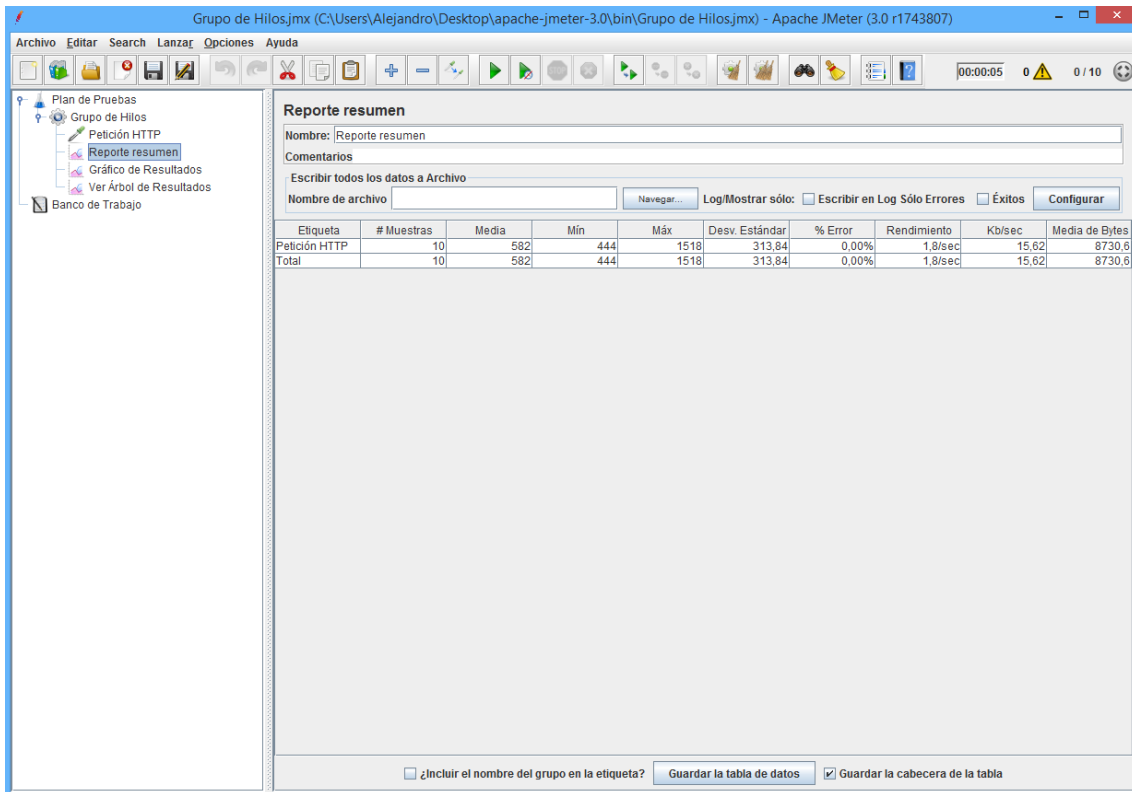


Figura 30 - Interfaz de JMeter

En primer lugar, a la izquierda nos encontramos con el plan de pruebas. En el añadimos los diversos elementos que compondrán las pruebas, como por ejemplo el grupo de hilos que simulará a los usuarios, elementos de configuración como el tipo de petición y diversos receptores para interpretar los resultados. En el panel de la derecha podemos observar los resultados de la prueba. En este caso con 10 usuarios, no se ha producido ningún error, y el servidor ha tenido un rendimiento de 1.8 peticiones servidas por segundo. También podemos observar el árbol de resultados para ver los detalles de las peticiones:

Ver Árbol de Resultados

Nombre:

Comentarios

Escribir todos los datos a Archivo

Nombre de archivo Log/Mostrar sólo: Escribir en Log Sólo Errores Éxitos

Buscar: Sensible a mayúsculas Expresión regular

Texto

- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP
- Petición HTTP

Resultado del Muestreador | Petición | Datos de Respuesta

Nombre del hilo: Grupo de Hilos 1-4
 Comienzo de muestra: 2016-08-25 17:46:54 CEST
 Tiempo de carga: 2005
 Connect Time: 2
 Latencia: 830
 Tamaño en bytes: 8731
 Headers size in bytes: 2073
 Body size in bytes: 6658
 Conteo de muestra: 1
 Conteo de error: 0
 Data type ("text"|"bin"): text
 Código de respuesta: 200
 Mensaje de respuesta: OK

Cabeceras de respuesta:
 HTTP/1.1 200 OK
 Date: Thu, 25 Aug 2016 15:46:55 GMT
 Server: Apache/2.4.18 (Win32) PHP/5.6.19
 X-Powered-By: PHP/5.6.19
 Cache-Control: no-cache
 Set-Cookie: XSRF-TOKEN=eyJpdil6ImRRRHFaQJUMTNZRVB6TmwrRkl2NHc9PSIsInZhbHVlIjoilShlcWpu
 expires=Thu, 25-Aug-2016 17:46:56 GMT; Max-Age=7200; path=/
 Set-Cookie: laravel_session=eyJpdil6Inhuc2k5YXprMWpwU1wvOHJyWkR6UXlBPT0iLCJ2YWx1ZSI6InY1dW5Kd0ZQO
 expires=Thu, 25-Aug-2016 17:46:56 GMT; Max-Age=7200; path=/; httponly
 Content-Length: 6282
 Keep-Alive: timeout=5, max=100
 Connection: Keep-Alive
 Content-Type: text/html; charset=UTF-8

Scroll automatically?

Figura 31 - Árbol de resultados de JMeter

Veamos a continuación los resultados de las pruebas llevadas a cabo:

# Usuarios	% Error	Rendimiento	Kb/sec
10	0,00 %	1,8/sec	15,62
25	4,00 %	3,8/sec	35,55
50	6,00 %	5,0/sec	47,81
200	25,50 %	5,9/sec	75,96
500	61,20 %	11,4/sec	61,13

Tabla 5 - Resultados de la prueba de estrés

Como puede observarse, a partir de los 25 usuarios aproximadamente empiezan a aparecer algunos errores. Veamos el árbol de resultados para el caso de 25 usuarios:

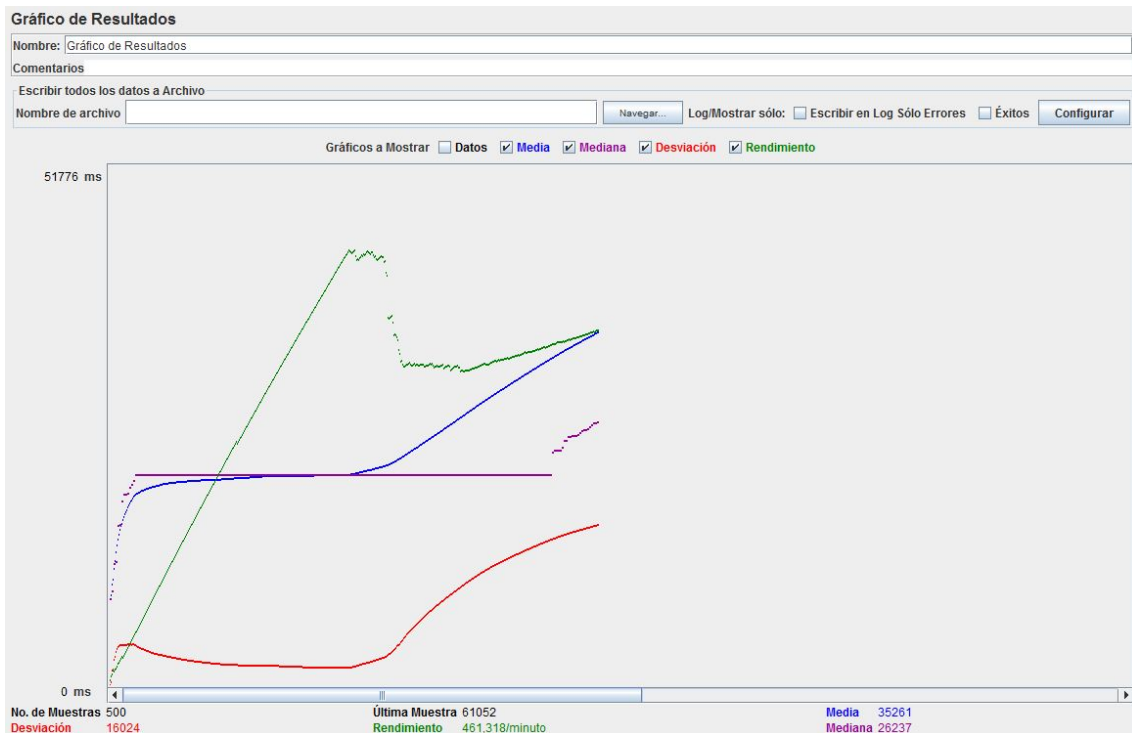


Figura 33 - Resultados de la prueba con 500 usuarios

La imagen anterior es de la prueba con los 500 usuarios. Podemos observar como el rendimiento (línea verde), se estanca alrededor de la mitad de la prueba, para posteriormente caer en picado y volver a recuperarse lentamente. En esta caída, la mayor parte de las peticiones no han podido ser respondidas y han sido descartadas. El punto en el que se produjo la caída corresponde con las peticiones de entre 250 y 300 usuarios concurrentemente.

A la vista de los resultados obtenidos, el servidor no debería presentar problemas a la hora de soportar varias peticiones concurrentes dentro del entorno de uso esperado. Por lo tanto, la estabilidad en el uso de la aplicación debería estar asegurada en todo momento.

7 Conclusiones

Se ha diseñado y desarrollado desde cero una herramienta para el aprendizaje del lenguaje C, la cual era necesaria para la Escuela Politécnica superior. Para ello se ha usado un framework de desarrollo usado en el entorno profesional, junto con las últimas tecnologías de desarrollo web. Todos los requisitos fijados inicialmente han sido implementados, por lo que finalmente se tiene una herramienta completamente funcional y lista para empezar a utilizar.

Esta aplicación servirá de apoyo a los estudiantes en su proceso de aprendizaje, pudiendo utilizarla, desde el primer día desde un navegador web, sin tener que preocuparse de instalar ningún tipo de entorno. Por otro lado, también será especialmente útil para los profesores, ya que podrán tener control y llevar un seguimiento de los alumnos de una forma sencilla y rápida.

En torno al trabajo futuro, se abren múltiples posibilidades. Se podría ampliar junto con el módulo de evaluación un módulo de edición de contenidos, esto es, una interfaz para editar los contenidos mostrados en el módulo docente: la documentación correspondiente a cada objetivo, los ejercicios, la documentación de ayuda y las cuestiones de opción múltiple. Este módulo permitiría al administrador del curso actualizar cómodamente los contenidos sin necesidad de tener un conocimiento detallado del proyecto.

También se podría hacer la aplicación colaborativa, por ejemplo, desarrollando funcionalidad para que los estudiantes puedan crear sus propias unidades o ejercicios, y compartirlos con los demás alumnos. Establecer un sistema de puntuaciones o recompensas en función de sus resultados, colaboraciones y/o aportaciones, que fomente el uso de la aplicación.

Sería también interesante, una vez se hubieran recogido los suficientes datos en la aplicación, añadir los datos reales de la evaluación y nota final de la asignatura, para posteriormente aplicar técnicas de machine-learning. Con estas técnicas, quizás podrían desarrollarse clasificadores, modelos predictivos o encontrar correlaciones que permitieran anticiparse, con suficiente tiempo, a posibles riesgos de suspenso por parte de los alumnos.

8 Referencias.

1. Codecademy (2011-2016). Learn to code interactively, for free. New York, USA. Recuperado de <https://www.codecademy.com>.
2. Codeschool (2011-2016). Learn by doing. Orlando, Florida, USA: Pluralsight Company. Recuperado de <https://www.codeschool.com>.
3. W3Schools (1999-2016). The world's largest web developer site. Norway: Refsnes Data. Recuperado de <http://www.w3schools.com>.
4. K. N. King. C Programming: A modern approach (second edition). W. W. Norton and Company, 2008.
5. By The Numbers: MOOCs in 2015 — Class Central. Class Central's MOOC Report. Retrieved 6 September 2016, from <https://www.class-central.com/report/moocs-2015-stats/>
6. Learn C - Free Interactive C Tutorial. (2016). Learn-c.org. Retrieved 6 September 2016, from <http://www.learn-c.org>
7. Online C Programming Course. (2016). Alison.com. Retrieved 6 September 2016, from <https://alison.com/courses/Diploma-in-Programming-in-C>
8. Learn C Programming: Start Coding in C Today on Udemy. (2016). Udemy. Retrieved 6 September 2016, from <https://www.udemy.com/c-programming-for-beginners/>
9. C programming.com - Learn C and C++ Programming - Cprogramming.com. (2016). Cprogramming.com. Retrieved 6 September 2016, from <http://www.cprogramming.com/>
10. HTML5. (2016). W3.org. Retrieved 6 September 2016, from <https://www.w3.org/TR/html5/>
11. CSS3 Introduction. (2016). W3schools.com. Retrieved 6 September 2016, from http://www.w3schools.com/css/css3_intro.asp

12. JavaScript. (2016). Es.wikipedia.org. Retrieved 6 September 2016, from <https://es.wikipedia.org/wiki/JavaScript>
13. Group, D. (2016). Welcome! - The Apache HTTP Server Project. Httpd.apache.org. Retrieved 6 September 2016, from <https://httpd.apache.org/>
14. MySQL. (2016). Mysql.com. Retrieved 6 September 2016, from <https://www.mysql.com/>
15. PHP: Manual de PHP - Manual. (2016). Secure.php.net. Retrieved 6 September 2016, from <https://secure.php.net/manual/es/index.php>
16. Otwell, T. (2016). Laravel - The PHP Framework For Web Artisans. Laravel.com. Retrieved 6 September 2016, from <https://laravel.com/>
17. Ace - The High Performance Code Editor for the Web. (2016). Ace.c9.io. Retrieved 6 September 2016, from <https://ace.c9.io/#nav=about>
18. Beautiful HTML5 JavaScript Charts | CanvasJS. (2016). CanvasJS. Retrieved 6 September 2016, from <http://canvasjs.com/>
19. Apache JMeter - Apache JMeter™. (2016). Jmeter.apache.org. Retrieved 6 September 2016, from <http://jmeter.apache.org/>

9 Anexo A – Índice del contenido docente

1. Buenas tardes
2. El peso ideal
3. El índice de masa
4. Un saludo con carácter
5. El desborde
6. Intervalos
7. Pares e impares
8. Interés compuesto
9. Velocidades
10. Conversor
11. Producto escalar
12. Determinante
13. Darse de alta
14. Fechas
15. Un juego de adivinación
16. Suma de números
17. Una matriz formateada
18. Otra suma de números
19. Suma de matrices
20. Agenda electrónica
21. Tres en raya

22. Complemento a 2
23. Hola dividido por dos
24. Conversor descompuesto
25. Media aritmética
26. Incrementar
27. El mayor
28. Números primos
29. Longitud de una cadena
30. Ordenar
31. Agenda dividida
32. La mediana
33. Contar caracteres
34. Contar palabras
35. Ventas
36. Agenda dinámica
37. Delimitadores
38. Media desordenada
39. Media en dos archivos
40. Media en tres archivos