

Aula interactiva para el aprendizaje a la programación

Urko Díaz Guardabrazos

Resumen— El proyecto de *Aula interactiva para el aprendizaje de la programación* consiste en el desarrollo de una aplicación web interactiva que facilite la docencia de diversas asignaturas de programación. Esta aplicación web permitirá que el profesor de una asignatura de programación pueda proponer ejercicios, para que los alumnos de la asignatura puedan resolverlos directamente a través de la interfaz web mediante un editor de texto y un compilador con el fin de ejecutar todo en un mismo entorno. El proyecto también debe permitir la interacción entre profesores y alumnos a tiempo real para que el profesor pueda ver cómo están trabajando los alumnos y poder mostrarlo a los demás alumnos en caso de necesitarlo. Para ello, es necesario que los alumnos puedan acceder a los ejercicios propuestos por el profesor directamente desde el aplicativo web sin necesidad de descargar ningún programa o archivos. Los alumnos podrán de este modo ver los ejercicios propuestos por el profesor, editar el código, guardarlo, corregir los ejercicios y subirlos al servidor comprimidos en formato zip o rar. El profesor, de esta manera, podrá ofrecer su feedback a los alumnos y ayudarles de una manera más sencilla a aprender a programar.

Palabras clave— Página web, Aula interactiva, Feedback, Compilador, Aplicación web, Moodle, plugin.

Abstract— The *Interactive classroom for learning programming* project consists in the development of an interactive web application that helps the teaching of various programming subjects. This web application will allow the teacher of a programming subject to propose exercises so that the students of the subject can solve them directly through the web interface through a text editor and a compiler in order to execute everything in the same environment. The project must also allow interaction between teachers and students in real time so that the teacher can see how students are working and show it to other students if needed. For this, it's necessary that students can access the exercises proposed by the teacher directly from the web application without having to download any program or files. Students can see the exercises proposed by the teacher, edit the code, save it, correct the exercises and upload them to the server compressed in zip or rar format. The teacher, in this way, can offer their feedback to students and help them in a simpler way to learn to program.

Index Terms— Website, Interactive classroom, Feedback, Compiler, Web application, Moodle, plugin.



1 INTRODUCCIÓN

POCO a poco, cada día más y más gente utiliza ordenadores de forma habitual. Esto hace que cada día más gente comience a introducirse en el mundo de la informática y con ello comienzan a ver cómo funcionan realmente los ordenadores. Las generaciones van pasando y ya es un hecho que es muy fácil tener un ordenador en casa o en la oficina. Los niños se crían con una tablet en sus manos y es totalmente normal comenzar a utilizar dispositivos inteligentes a una edad muy corta.

Esto conlleva a que cada vez se esté más familiarizado con los ordenadores y la informática y, de hecho, la informática es cada día más necesaria.

Tal es así, que cada día más gente quiere o necesita aprender a programar para tener un abanico de oportunidades más amplio de cara a su futuro. Y a la necesidad de aprender a programar hay que añadirle la necesidad de enseñar a programar.

Es por ello que se ha llevado a cabo este proyecto llamado *Aula interactiva para el aprendizaje a la programación*.

Con este proyecto, se busca facilitar la manera en la que se enseña a programar ya que no es una tarea sencilla. Se busca que el profesor pueda, de una forma más fácil, enseñar y ayudar a sus alumnos desde una única plataforma y sin necesidad de descargar archivos o programas.

Esta plataforma permite a los profesores crear distintos cursos para distintas asignaturas en los que pueden ir añadiendo distintas sesiones de laboratorio con diversos ejercicios de programación que el profesor desee

- E-mail de contacto: urko.diazg@e-campus.uab.cat
- Mención realizada: Ingeniería del Software.
- Trabajo tutorizado por: Ernest Valveny Llobet (Departamento de Ciencias de la Computación)
- Curso 2017/18

plantear a sus alumnos.

Por otro lado, permite a los alumnos acceder a un entorno en el que el propio alumno puede ver los cursos que están activos y en ellos las diversas sesiones y ejercicios planteados por los profesores.

2 OBJETIVOS

El objetivo principal del proyecto consiste en elaborar un aula interactiva con carácter de aplicación web dividida en dos partes o sub-aplicaciones (una enfocada a los alumnos y la otra enfocada a los profesores que, conjuntamente, sirvan como un único aplicativo web conectado entre sí) que permitan editar, compilar y ejecutar código, y que este se pueda enviar al profesor como respuesta al ejercicio planteado. La aplicación web debe corregir el ejercicio del alumno y devolverle la nota resultante del ejercicio entregado.

Al ser un proyecto muy extenso y centrado en ser desarrollado entre dos personas, este proyecto en concreto se centrará en la parte del profesor en el que se permitirá la inserción de ejercicios y la visualización de los códigos generados por los alumnos.

Tras realizar un estudio del entorno para ver qué otros aplicativos web hay actualmente, y después de mantener una reunión con el tutor del TFG, se ha decidido desarrollar el aplicativo web utilizando la herramienta Moodle. Esta herramienta está concebida para ayudar a los profesores a crear comunidades de aprendizaje en línea. Dada su utilización, se ha creído conveniente utilizar esta herramienta, ya que está pensada para facilitar el trabajo tanto a alumnos como a profesores en las labores de enseñanza, haciendo que el aplicativo web resulte fácil de utilizar y muy intuitivo.

Esta herramienta es de gran utilidad en el ámbito educativo, ya que permite a los profesores la gestión de cursos virtuales para sus alumnos, o el uso de un espacio en línea que sirva de ayuda para los alumnos a la hora de realizar diversas prácticas o ejercicios de programación.

Por otro lado, esta herramienta permite fácilmente el acceso al aula interactiva a través de un teléfono móvil, con lo que tanto profesores como alumnos pueden acceder al entorno desde cualquier lugar.

El hecho de trabajar con la herramienta Moodle, nos permite agregarle distintos plugins que facilitan la tarea del profesor, como por ejemplo el VPL, que sirve para poder programar, compilar y ejecutar un código desde la propia aplicación web. Más adelante se detallará el uso de este plugin.

3 ESTADO DEL ARTE

3.1 Situación actual

Actualmente, no hay ninguna aplicación web en el mercado que permita editar, ejecutar y guardar los mismos archivos de forma online y a tiempo real, con lo que con este aplicativo web podremos mejorar el funcionamiento de las prácticas de asignaturas de programación para que los alumnos puedan aprender de una forma más rápida y personalizada.

Sí que podemos encontrar herramientas^[4] que hacen las diferentes funciones por separado, pero ninguno que englobe las funcionalidades descritas anteriormente. Por otro lado, estas aplicaciones suelen ser de pago o necesitan que se verifique que se es alumno o profesor y ofrecen sus herramientas de manera gratuita por un tiempo limitado. A continuación detallamos algunas de estas herramientas:

- CodeHS^[13]: CodeHS es una plataforma de enseñanza integral para ayudar a las escuelas a enseñar informática. Ofrecen currículo basado en la web, herramientas y recursos para docentes, y desarrollo profesional. Esta aplicación web ofrece 2 meses gratuitos y a partir del 3er mes tiene un coste de 75\$ mensuales.

- Clip2net^[14]: Esta aplicación sirve tanto para web como para escritorio, Android e iOS. Es una herramienta que permite compartir ficheros e imágenes de manera sencilla entre distintos dispositivos. Ofrece 3 planes distintos con distintas funcionalidades. El plan *Base* es gratuito, el plan *Lite* tiene un coste de 11.95\$ anuales y el plan *Pro* tiene un coste de 29.95\$ anuales.

- Codeshare.io^[15]: Esta aplicación web es una herramienta para compartir código con otros usuarios a tiempo real. Es una herramienta gratuita.

- Programmr^[16]: Programmr es una herramienta que ofrece distintos cursos y retos para aprender a programar en diferentes lenguajes de programación.

- Codepad.org^[17]: Esta aplicación web ofrece un compilador que permite pegar código y ejecutarlo en diferentes lenguajes.

A partir de esta información, se nos ha asignado un proyecto que consiste en englobar dichas funcionalidades para hacer más fácil la forma de enseñar a programar de los profesores, ya que podrán ver lo que hacen sus

alumnos a tiempo real y ofrecerles su feedback para que los alumnos puedan aprender de mejor forma.

Este proyecto lo estamos realizando dos alumnos (David Ortiz Gallardo y yo) aunque lo vamos a realizar de manera individual, centrándose David en la parte enfocada a los alumnos y yo realizando la parte enfocada a los profesores.

3.2 Descripción del problema

Debido a que no hay ninguna aplicación web que permita realizar los ejercicios de forma interactiva sin tener que descargarlos, hay que empezar todo desde cero, sin ninguna base.

El profesor nos ha facilitado varias herramientas online (mencionadas en el punto 3.1) que entre todas ellas podrían hacer el funcionamiento que se nos pide en este proyecto, pero el proyecto se centra en que desde un mismo aplicativo web se puedan realizar todas las funcionalidades descritas y a tiempo real.

En numerosas ocasiones, el profesor quiere ver las ejecuciones de los alumnos con el fin de extraer información de sus errores, es por eso que esta aula interactiva debería disponer únicamente de un explorador web para poder ejecutar los ejercicios y permitir subir al servidor los ejercicios solucionados.

3.3 Alcance del proyecto

Con este proyecto, se intentará realizar una serie de objetivos que nos permitirán establecer un punto de partida y finalmente unas conclusiones de todo nuestro proceso.

Durante los próximos 3 meses de realización del proyecto, nuestro principal objetivo es cumplir y asegurar los requerimientos que nos ha transmitido el profesor (recopilando la información que nos pide), para posteriormente representarlas en una aplicación web para alumnos y profesores.

3.4 Motivación

Viendo la situación actual de las aplicaciones web y sabiendo que no hay ninguno que sea como el que se quiere implementar, se nos ha despertado una motivación para resolverlo.

Es necesaria una aplicación web que reúna estas características para facilitar las clases de prácticas, ya que de esta manera los alumnos, mediante sus ordenadores, pueden practicar en clase los problemas planteados por el profesor al momento y ejecutarlos para saber su resulta-

do.

4 METODOLOGÍA Y PLANIFICACIÓN

4.1 Metodología

La metodología que se utilizará para llevar a cabo este proyecto será la metodología incremental. El modelo incremental combina elementos del modelo en cascada con la filosofía interactiva de construcción de prototipos. Se basa en la filosofía de construir incrementando las funcionalidades del programa. Este modelo aplica secuencias lineales de forma escalonada mientras progresa el tiempo en el calendario. Cada secuencia lineal produce un incremento del software.

Al ser un proyecto que debe ser realizado por dos personas, es muy importante que exista una comunicación entre ambas partes para poder realizar un desarrollo óptimo, ya que hay muchas cosas que se deberán estudiar en común para poder realizar la transmisión de ficheros y conexión a tiempo real.

La metodología se dividirá en las cinco fases o sub-objetivos:

- Estudio del entorno
- Análisis de la solución
- Diseño de la solución
- Desarrollo de la solución
- Testing de la aplicación web.

4.2 Planificación

En cuanto a la planificación, para llevar a cabo el desarrollo del proyecto que se describe en este informe, se realizó una planificación basada en 3 fases divididas en un tiempo aproximado de 1 mes para cada fase.

Gracias a que se decidió que el diseño del aplicativo web estuviera desarrollado con Moodle, esto nos ahorró mucho tiempo de desarrollo de front-end, centrándonos más en que el aplicativo funcionase correctamente y sin errores y permitiera hacer más amenas, entretenidas y fáciles las sesiones de laboratorio de las asignaturas de programación.

4.2.1 Fase de inicio

La fase de inicio del proyecto de *Aula interactiva para el aprendizaje a la programación* consta de las tareas que pueden verse en la tabla 1 - fase de inicio.

| | |
|---|--|
| ✓ | Reunión inicial entre el tutor Ernest Valveny, David Ortiz y yo. |
| ✓ | Búsqueda de información. |

| | |
|---|-------------------------------------|
| ✓ | Estudio del entorno. |
| ✓ | Decidir los requisitos del sistema. |
| ✓ | Realizar la planificación. |
| ✓ | Análisis de la solución. |
| ✓ | Entrega del informe inicial. |

Tabla 1 - Fase de inicio

| | |
|---|--|
| ✓ | Reunión de seguimiento 4. |
| ✓ | Entrega de la propuesta del informe final. |
| ✓ | Entrega de la documentación del proyecto. |
| ✓ | Entrega del producto. |
| ✓ | Defensa del proyecto. |

Tabla 4 - hitos de la fase de finalización

4.2.2 Fase de desarrollo

La fase de desarrollo del proyecto de *Aula interactiva para el aprendizaje a la programación* consta de hitos y tareas. Los hitos que se han llevado a cabo en la fase de desarrollo del proyecto pueden verse en la tabla 2 - hitos de la fase de desarrollo.

| | |
|---|------------------------------------|
| ✓ | Reunión de seguimiento 1. |
| ✓ | Escoger metodología del proyecto. |
| ✓ | Entrega del informe de progreso 1. |
| ✓ | Reunión de seguimiento 2. |
| ✓ | Reunión entre David Ortiz y yo. |
| ✓ | Entrega del informe de progreso 2 |
| ✓ | Reunión de seguimiento 3. |

Tabla 2 - hitos de la fase de desarrollo

A continuación se muestran las tareas realizadas que completan esta fase de desarrollo. Estas tareas pueden verse en la tabla 3 - tareas de la fase de desarrollo.

| | |
|---|--|
| ✓ | Realizar documentación. |
| ✓ | Hacer un primer prototipo en local. |
| ✓ | Corrección del informe inicial. |
| ✓ | Redactar el informe de progreso 1. |
| ✓ | Subir el prototipo al servidor. |
| ✓ | Instalar en el servidor el plugin para Moodle de VPL. |
| ✓ | Instalar PHP y MySQL en el servidor. |
| ✓ | Corrección del informe de progreso 1. |
| ✓ | Crear perfiles de administrador, profesor y alumno. |
| ✓ | Redactar el informe de progreso 2. |
| ✓ | Subir ejercicios a la base de datos de Moodle para poder crear ejercicios a partir de ellos. |
| ✓ | Testeo del aplicativo. |
| ✓ | Corrección del informe de progreso 2. |
| ✗ | Desarrollar plugin para facilitar la creación de sesiones. |
| ✓ | Corrección de errores. |

Tabla 3 - tareas de la fase de desarrollo

4.2.3 Fase de finalización

La fase de finalización del proyecto de *Aula interactiva para el aprendizaje a la programación*, también consta de hitos y tareas que pueden verse en las tablas 4 y 5 que se muestran a continuación.

| | |
|---|---|
| ✓ | Revisar y corregir la documentación del proyecto. |
| ✓ | Corrección de los últimos errores del proyecto. |
| ✓ | Preparar presentación en Prezi. |

Tabla 5 - Tareas de la fase de finalización

5 REQUERIMIENTOS

En la sección de requerimientos se mostrarán tanto los requerimientos funcionales como los no funcionales del aula interactiva para el aprendizaje a la programación.

5.1 Requerimientos Funcionales

Los requerimientos funcionales del aula interactiva para el aprendizaje a la programación pueden verse en la *tabla 6*.

| |
|---|
| R.F. 1. El aplicativo web deberá permitir crear diferentes cursos y sesiones. |
| R.F. 2. El aplicativo web deberá permitir crear problemas o ejercicios en los que se permita ejecutar código mediante el plugin de VPL. |
| R.F. 3. El aplicativo web deberá permitir mostrar u ocultar problemas a los alumnos. |
| R.F. 4. El aplicativo web deberá permitir establecer ciertos criterios para la entrega de ejercicios tales como: Fecha de entrega, límite de entregas, ficheros requeridos, evaluación automática, etc. |
| R.F. 5. El aplicativo web deberá permitir probar las actividades creadas. |
| R.F. 6. El aplicativo web deberá permitir descargar las entregas de todos los alumnos. |
| R.F. 7. El aplicativo web deberá permitir evaluar automáticamente las entregas de los alumnos. |
| R.F. 8. El aplicativo web deberá permitir mostrar a los alumnos en tiempo real lo que el profesor va escribiendo. |

Tabla 6 - Requerimientos funcionales

Algunos de estos requerimientos funcionales pueden relacionarse con algunos objetivos del aplicativo web.

Por ejemplo, el requerimiento funcional 2 puede relacionarse con el objetivo de ejecutar y compilar código desde el propio servidor sin necesidad de descargarse programas o archivos adicionales.

5.2 Requerimientos no funcionales

A continuación, en la tabla 7 se detallan los requerimientos no funcionales del aplicativo web.

| |
|--|
| R.N.F. 1. El aplicativo web deberá tener el diseño de Moodle |
| R.N.F. 2. El aplicativo web deberá tener un diseño fácil e intuitivo tanto para los alumnos como para los profesores. |
| R.N.F. 3. Los ejercicios creados por el profesor deberán estar almacenados en una base de datos para poder ser accedidos cuándo el profesor lo desee. |
| R.N.F. 4. El aplicativo web deberá ser compatible con cualquier navegador web. |
| R.N.F. 5. El aplicativo web deberá ser multiplataforma, es decir, que se podrá visualizar desde cualquier dispositivo y/o navegador. |
| R.N.F. 6. El aplicativo web deberá poder visualizarse con la aplicación de Moodle para teléfonos móviles. |

Tabla 7 - Requerimientos no funcionales

Algunos de estos requerimientos no funcionales pueden relacionarse con algunos objetivos del aplicativo web.

Por ejemplo, el requerimiento no funcional 2 puede relacionarse con el objetivo de ser un aula interactiva con un diseño intuitivo para permitir un mejor aprendizaje.

6 DESARROLLO

Como se ha mencionado previamente, el software se ha realizado con la herramienta Moodle. Para la utilización de esta herramienta, se ha procedido a instalarla en un servidor prestado por la universidad el cual se ha configurado de manera que nos permita realizar el desarrollo del software sobre él.

Se ha instalado la herramienta Moodle^[6] 3.4.2+ para el desarrollo del aula interactiva y para la base de datos se ha instalado MySQL y PHP 7.0. De esta manera estaremos utilizando las versiones más actualizadas de estas herramientas.

Una vez configuradas las herramientas y después de hacer algún que otro ajuste para que se nos permita subir archivos a la herramienta Moodle^[7], se ha procedido a instalar el plugin VPL o Virtual Programming Lab.

VPL para Moodle permite tanto a alumnos como a profesores tener un espacio en el aula interactiva para poder escribir, compilar y ejecutar código así como calificar las entregas de los alumnos automáticamente.

Este plugin ha sido desarrollado por un profesor de la Universidad de Las Palmas de Gran Canaria y es de gran utilidad para aprender a programar, ya que permite ejecutar códigos en diferentes lenguajes de programación tales como C, C++, C#, Java, Pascal, PHP, Python, Ruby, SQL, etc.

Para que este plugin pueda ejecutar código en el mismo navegador sin necesidad de descargar ningún programa, lo que hace es desde el propio navegador convertir el código introducido por el alumno a AJAX en formato json y enviarlo al servidor de Moodle. Después cifra la información y lo envía a otro servidor desde el que se ejecuta el código.

Gracias a las facilidades de uso de este plugin, se ha podido llevar a cabo el proyecto planteado en el cual el profesor puede crear diferentes sesiones y añadir los ejercicios que desea que los alumnos realicen.

Como podemos ver en la ilustración 1, el profesor puede acceder a los distintos cursos que ha creado, crear más cursos, modificarlos, editarlos, etc.

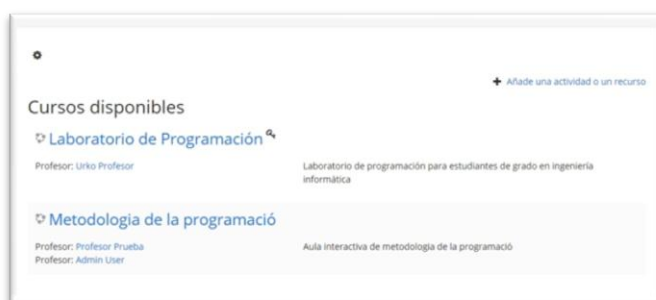


Ilustración 1 - Creación de cursos

Tal y como se muestra en la ilustración 2, se han creado varias sesiones simulando ser prácticas de laboratorio en el que se han ido añadiendo ejercicios visibles para los alumnos u ocultos, según las preferencias del profesor, y que se pueden ir haciendo visibles para los alumnos en el momento que el profesor lo decida.

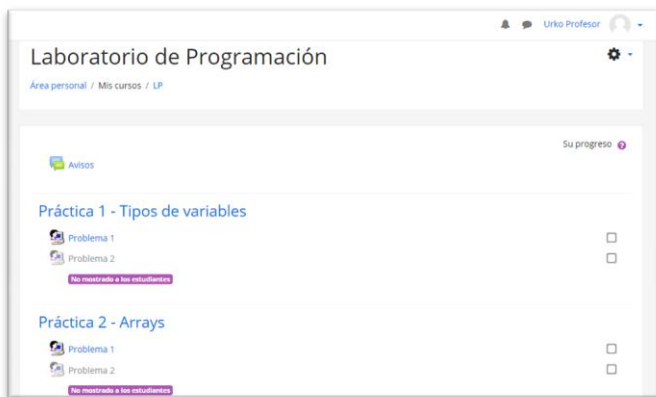


Ilustración 2 - Sesiones y problemas

Estos ejercicios, están ya disponibles para que los alumnos vean el esqueleto del código y puedan editarlo. Como se muestra en la ilustración 3, podemos ver los requisitos fijados por el profesor en cuanto a la fecha de entrega, ficheros requeridos y si se prefiere que el propio plugin de VPL junto con el código `vpl_evaluate.sh` proporcionado por el tutor Ernest Valveny nos autoevalúe el código entregado por el alumno.

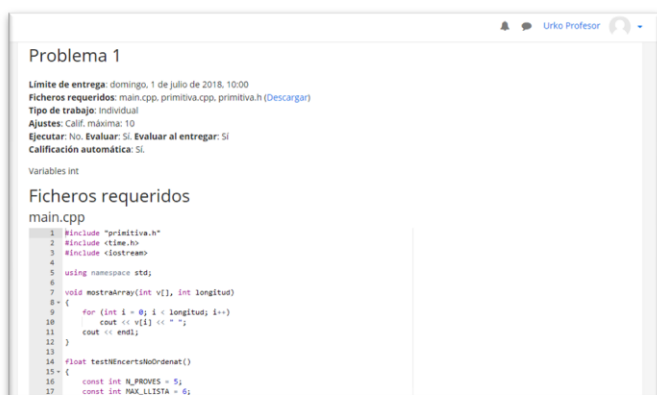


Ilustración 3 - Requisitos de la entrega

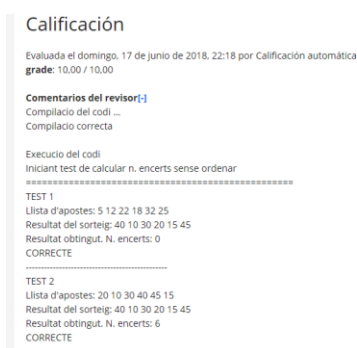


Ilustración 4 - Autoevaluación

Si se selecciona que el propio programa autoevalúe las entregas de los alumnos, se puede prefijar esta preferencia en las opciones de entrega del problema en

cuestión. Esto resultará en unas comprobaciones tal y como se muestra en la ilustración 4 en las que el propio programa ejecuta el código entregado por el alumno, comprueba las diferentes salidas y devuelve una nota.

Esto es muy útil tanto para el alumnado como para el profesor, ya que el alumno puede saber si está realizando un buen trabajo o no, y así le permite ver en qué puntos del ejercicio está fallando. Por otro lado, esto ayuda al profesor a la hora de realizar las evaluaciones, ya que siendo automáticas no tiene que corregir los ejercicios de los alumnos uno por uno, pudiendo más adelante descargarse solo los ejercicios entregados por los alumnos que no han superado la nota mínima del ejercicio.

Por otro lado, la herramienta permite al profesor añadir ejercicios o sesiones ocultas para los alumnos y tenerlos ya creados para en un momento dado darles acceso a los alumnos o mostrarles otros ejercicios diferentes.

Esto también ayuda a la hora de establecer qué ejercicios quiere el profesor que realicen los alumnos y darles acceso a los ejercicios ocultos para que puedan seguir con las sesiones de laboratorio o aprender todavía más, ya que se pueden configurar las sesiones para que si un alumno ha superado ciertos ejercicios, se le desbloquee en otros ejercicios para poder seguir practicando o aprendiendo.

En cierto punto, el profesor puede necesitar editar el código para mostrárselo a los alumnos y el aplicativo web también le permite realizar esta acción, abriéndose el editor de texto en cualquiera de los ejercicios que el profesor lo desee, tal y como se muestra en la ilustración 5.



Ilustración 5 - Editar actividad

Para poder llegar a este punto, se ha procedido a crear dos perfiles a parte del perfil administrador (uno con rol de profesor y otro con rol de alumno) para poder ir testeando que todo lo que se va implementando funciona a la perfección. A pesar de que la herramienta Mo-

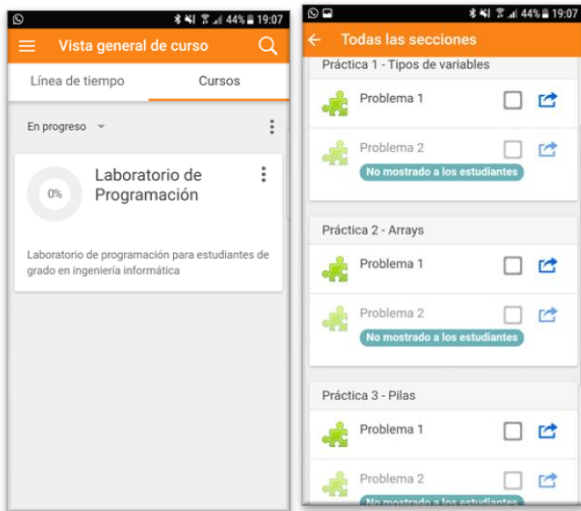
odle nos permite probar las actividades propuestas a los alumnos, se ha preferido crear un perfil de alumno para así tener más claro qué es lo que se está mostrando a los alumnos.

Por otro lado, esto también nos ha permitido realizar entregas con el perfil de alumno y probar que el profesor puede descargar correctamente todos los archivos entregados por cada alumno. La herramienta Moodle, además, nos permite descargar cualquier entrega realizada por el alumno aunque ésta no sea la entrega final del alumno, ya que quedan registrados y guardados todos los ficheros subidos por el alumno.

Para hacer esta labor un poco más sencilla, la herramienta de Moodle nos permite descargar todas las entregas de todos los alumnos a la vez, separadas en carpetas pero todas en un mismo archivo zip, ahorrando trabajo al profesor de tener que descargar una a una cada entrega.

El equipo de desarrollo de Moodle ha hecho un gran trabajo a la hora de adaptar su herramienta a cualquier resolución y dispositivo. Tanto es así, que han creado una aplicación para móviles desde la que es impresionantemente fácil acceder a los cursos de Moodle en los que estemos matriculados, tanto si somos profesor como alumno. Basta con descargarla, escribir la URL de nuestro curso (en nuestro caso aulainteractiva.uab.cat/moodle) e iniciar sesión.

Como se puede comprobar en las ilustraciones 6 y 7, la aplicación nos permite ver todos los cursos en los que estamos matriculados, sus sesiones y sus problemas.



Ilustraciones 6 y 7 - Aplicación Moodle para móviles

Sin embargo, dado a que nosotros estamos trabajando en un curso con la finalidad de aprender o enseñar

a programar, necesitamos trabajar con el plugin VPL, el cual no está integrado en la aplicación. Este hecho no es un problema, ya que la propia aplicación detecta que necesita un plugin que no está disponible y nos da la opción de abrir el navegador para poder ver el curso correctamente.

Como comentaba anteriormente, el equipo de desarrollo ha realizado un gran trabajo y el navegador del teléfono móvil puede abrir sin ningún problema el curso que deseamos ver con el plugin de VPL, incluso nos permite realizar las mismas funciones que el navegador para PC.

7 DIAGRAMA DE CASOS DE USO

Para poder ejemplificar la manera en la que funciona el apartado de profesor del proyecto *Aula interactiva para el aprendizaje a la programación*, se hará a través del diagrama de casos de uso que se muestra a continuación.

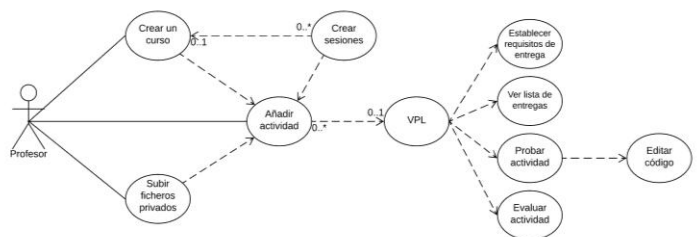


Ilustración 8 - Diagrama de casos de uso

8 RESULTADOS OBTENIDOS

Llegados a este punto, cabe destacar que se ha logrado el objetivo principal de este proyecto, que es el de crear un aula interactiva que permita la inserción, edición y compilación de código para la enseñanza a programar.

Al ser un aula interactiva basada en Moodle, permite que el acceso a los cursos y diferentes sesiones sea relativamente fácil, sin necesidad de descargar ningún programa externo y todo ejecutado desde el propio navegador, permitiendo incluso el acceso desde el teléfono móvil.

Sin embargo, no se ha podido obtener un aula interactiva a tiempo real en la que el profesor vaya editando código y a los alumnos les aparezca esta edición de texto al momento en sus pantallas. Por ello, nos pusimos en contacto con los desarrolladores del plugin de VPL para preguntarles por una solución a este problema. Nos

ayudaron mucho, ya que nos comentaron que no era posible hacerlo a tiempo real, pero que sí era posible que el profesor edite el código y lo resubiera.

Para ello, el profesor puede editar el código mientras se lo muestra a los alumnos y después debe escoger la opción de descargar como zip. Una vez descargado el fichero puede resubirlo y lo único que deben hacer los alumnos es hacer click sobre el botón que dice *Restablecer ficheros* indicado con el símbolo:



Ilustración 9 - Botón "Restablecer ficheros"

Por otro lado, tampoco ha sido posible el realizar el desarrollo de un plugin instalable sobre Moodle para permitir crear sesiones más fácilmente. Se estudiará realizar el desarrollo más adelante para hacer más fácil la creación de sesiones por parte del profesorado.

A continuación se muestra una tabla con los requisitos tanto funcionales como no funcionales que se han mencionado en el apartado 5, y si ha sido posible llevarlo a cabo o no:

| | | | |
|---|---------|---|-----------|
| ✓ | R.F. 1. | ✓ | R.N.F. 1. |
| ✓ | R.F. 2. | ✓ | R.N.F. 2. |
| ✓ | R.F. 3. | ✓ | R.N.F. 3. |
| ✓ | R.F. 4. | ✓ | R.N.F. 4. |
| ✓ | R.F. 5. | ✓ | R.N.F. 5. |
| ✓ | R.F. 6. | ✓ | R.N.F. 6. |
| ✓ | R.F. 7. | | |
| ✗ | R.F. 8. | | |

Tablas 8 y 9 - Valoración de requisitos

9 PRUEBAS

Con tal de testear las diferentes funcionalidades de la plataforma, se han realizado diferentes tipos de test.

Como se ha mencionado anteriormente, se han creado perfiles tanto de profesor como de alumno para comprobar el correcto funcionamiento y se han realizado tests manuales de caja negra con tal de evaluar los requisitos funcionales a partir de casos de pruebas creados mediante los diferentes casos de uso.

10 CONCLUSIÓN

Desde un principio, se puede observar que este proyecto se trata de un proyecto de grandes dimensiones,

dividido en dos sub-proyectos. Uno centrado en la parte del alumno, y otro centrado en la parte del profesorado.

Se trata de un proyecto que tiene la intención de hacer más fácil la labor del profesor a la hora de llevar a cabo una sesión de laboratorio, y al alumno a la hora de aprender a programar.

Al principio, no se sabía cómo se podía llegar a ayudar tanto al profesorado como al alumnado a realizar sus labores, pero con el desarrollo del proyecto basado en Moodle esta tarea ha sido mucho más sencilla. Finalmente se ha conseguido llegar a buen puerto y el resultado es satisfactorio, ya que como alumno, he de decir que no es fácil aprender a programar, sobre todo si nunca se ha programado nada.

Sí que es cierto que hay algunas asignaturas en las que el aprendizaje es más costoso y otras en la que es mucho más sencillo avanzar y aprender. Sin embargo, con aplicativos web como este, se trata de facilitar el modo en el que se enseña a programar para que el alumnado tenga mayores facilidades a la hora de aprender, que, al fin y al cabo, es de lo que se trata.

Este aplicativo web permite ver el código del esqueleto, editarlo, compilarlo, entregarlo y evaluarlo desde el mismo entorno y sin esperas, lo que facilita mucho al alumnado realizar las prácticas y saber en qué ha fallado o ha tenido éxito.

Por otro lado, hace mucho más fácil la planificación de sesiones, visualizar las entregas de los alumnos, ayudar a los que se quedan más atrás y dar la oportunidad de seguir enseñando a los alumnos más avanzados, ya que permite hacer un seguimiento de cada alumno inscrito en cada curso.

Al haber realizado este proyecto, como se ha mencionado anteriormente, se entiende que la propuesta de aula interactiva flexible cumple con los requisitos establecidos del proyecto y funciona correctamente.

En el aspecto personal, con este trabajo he podido poner en práctica muchos aspectos y metodologías aprendidas a lo largo de mi estancia en la universidad, en especial sobre la mención de ingeniería del software.

AGRADECIMIENTOS

Me gustaría agradecer a mi familia, ya que sin su apoyo no habría conseguido llegar hasta aquí.

A mi novia Marina Sánchez, que me ha ofrecido su apoyo durante toda la carrera en los buenos y en los

malos momentos y me ha hecho crecer como persona.

A mi compañero de proyecto, David Ortiz, ya que hemos podido trabajar en equipo y ha sido muy enriquecedor a nivel personal.

A mis compañeros de piso, que tan buenos momentos me han hecho pasar.

A todos los amigos que he hecho a lo largo de mis años en la universidad que tan influyentes han sido para mí.

A todos los compañeros de las practicas de las distintas asignaturas de la carrera que de todos ellos he aprendido algo.

A mi tutor del TFG Ernest Valveny que sin sus consejos y ayuda no podría haber llevado a cabo este proyecto.

A Jordi Pons Aróztegui, por su paciencia con todos los alumnos y por todos los consejos y ayuda que me ha ofrecido durante toda mi carrera.

BIBLIOGRAFÍA

- [1] Es.wikipedia.org. (2018). *Aplicación web*. [online] https://es.wikipedia.org/wiki/Aplicación_web
- [2] Es.wikipedia.org. (2018). *Compilador*. [online] <https://es.wikipedia.org/wiki/Compilador>
- [3] Vpl.dis.ulpgc.es. (2018). *VPL - Virtual Programming Lab - About*. [online] <http://vpl.dis.ulpgc.es/index.php/about>
- [4] Arnemertz.github.io. (2018). *List of Online C++ Compilers by arnemertz*. [online] <https://arnemertz.github.io/online-compilers/>
- [5] Isw-udistrital.blogspot.com.es. (2018). *Modelo Incremental*. [online] <http://isw-distrital.blogspot.com.es/2012/09/ingenieria-de-software-i.html>
- [6] Docs.moodle.org. (2018). *Instalación de Moodle - MoodleDocs*. [online] https://docs.moodle.org/all/es/Instalación_de_Moodle
- [7] Clouding Knowledge Base. (2018). *Aumentar el límite de upload en PHP*. [online] <https://clouding.io/kb/aumentar-el-limite-de-upload-en-php/>
- [8] Es.wikipedia.org. (2018). *Moodle*. [online] <https://es.wikipedia.org/wiki/Moodle>
- [9] Vpl.dis.ulpgc.es. (2018). *VPL - Virtual Programming Lab - Home*. [online] <http://vpl.dis.ulpgc.es/>
- [10] Vpl.dis.ulpgc.es. (2018). *VPL - Virtual Programming Lab - Support*. [online] <http://vpl.dis.ulpgc.es/index.php/support>
- [11] En.wikipedia.org. (2018). *VPL Research*. [online] https://en.wikipedia.org/wiki/VPL_Research
- [12] Science.smith.edu. (2018). *Moodle VPL Tutorials - dftwiki*. [online] http://www.science.smith.edu/dftwiki/index.php/Moodle_VPL_Tutorials
- [13] Codehs.com. (2018). *What is CodeHS? | CodeHS*. [online] Available at: <https://codehs.com/info/>
- [14] Clip2net.com. (2018). *Clip2Net - screen capture tool for Windows, Android, iPad, Mac, Linux*. [online] Available at: <https://clip2net.com/en/>
- [15] Codeshare.io. (2018). *Codeshare - Share code in real-time with developers in your browser*. [online] Available at: <https://codeshare.io/>
- [16] Programmrr.com. (2018). *Programmrr | Your Online Code Lab*. [online] Available at: <http://www.programmrr.com/>
- [17] Codepad.org. (2018). *codepad*. [online] Available at: <http://codepad.org/>

ANEXOS

A.1 - Manual Moodle para el profesor

Documentación adicional sobre el funcionamiento de Moodle para el profesorado.

A.2 - Ejemplos código C++

Ejemplos de código para el plugin VPL para poder realizar un testeo a fondo de la aplicación.

A.3 - Código Plugin

Archivos del código del desarrollo del plugin para crear varias sesiones a la vez con sus respectivos ejercicios.