

Desarrollo de un front-end para DuoCode

Julián F. Calleja da Silva

Johana Gabriela Ferreira Yagua

José Carlos Valera Villalba

Grado en Ingeniería Informática

Facultad de Informática

Departamento de Sistemas Informáticos y Computación



Trabajo fin de grado

Dirigida por
Enrique Martín
Adrián Riesco

Madrid, 2015

Abstract

Resumen

Índice

1. Introducción con ejemplos	1
2. Hola	1
2.1. Introducción	1
3. Introducción	2
3.1. Investigación de campos	2
3.2. Revisión del estado del arte	2
3.3. Influencias tecnológicas	3
3.4. Propuestas y objetivos	3
A. Primera sección del apéndice	6

1. Introducción con ejemplos

Hola **negrita** *cursiva*

Tres

- Primera cosa
- Segunda cosa

1.

2.

Cosa 1. Esto que vamos a comentar ahora es algo muy importante porque blabla Esto que vamos a comentar ahora es algo muy importante porque blabla

Cosa2

Ejemplo 1

hola	<i>hola</i>	<i>hola</i>
<i>adios</i>	<i>adios</i>	<i>adios</i>

2. Hola

2.1. Introducción

Para la sección 2.1 usaremos el libro [13]

```
if (condicion){  
    int x = 0;  
}  
else{  
    int x = 1;  
}
```

3. Introducción

Este proyecto empezó con el objetivo de permitir a los alumnos que lo desarrollan profundizar en tecnologías y paradigmas que no se aprenden a lo largo del grado o que se dan como meras referencias. Todo ello sin olvidarnos de los valiosos conocimientos ya afianzados que seguro serán de utilidad. Queríamos hacer algo que ayude a la vida de las personas y que se pueda integrar con facilidad con la manera en que actualmente se interactúa con la tecnología.

3.1. Investigación de campos

Antes de delimitar aspectos más técnicos, debatimos sobre donde podríamos centrar nuestros esfuerzos. Nos dimos cuenta de que la educación podría ser un área muy interesante. Esta es algo que siempre ha estado presente en nuestra sociedad y la actual crisis no hace sino acentuar esto. Ante la falta de empleo o la perspectiva de que esto pueda suceder, las personas naturalmente quieren mejorar sus capacidades. Y mejorar la formación propia es una de las maneras más habituales de hacerlo. Nos planteamos el hecho de que las tasas universitarias han subido y que esto puede producir que las personas busquen métodos alternativos de enseñanza más baratos, como pueden ser sistemas de aprendizaje online. Por todo ello, por ser un campo en auge y a la vez muy estable, decidimos centrarnos en el 'E-learning', o sector educativo a través de tecnologías digitales, para desarrollar nuestro proyecto.

3.2. Revisión del estado del arte

En la actualidad hay numerosas maneras de aprender online. Para empezar existen recursos típicos, donde el que quiere aprender lee y realiza poca interacción. El líder por antonomasia de esta modalidad es Wikipedia[12] (o libros/bibliotecas de manera presencial). Ante dudas concretas hay sitios de preguntas y respuestas como el archiconocido para los programadores Stack Overflow[10]. En las universidades y otros centros educativos también se aprende y muchas cuentan con plataformas digitales como Moodle[7], que se suelen usar para publicar apuntes de las diferentes asignaturas. También existen cursos online, los llamados MOOC (acrónimo en inglés de massive open online course) donde puedes aprender a distancia temas nuevos. Estos requieren muchos recursos, y que se suba mucho material (vídeos, explicaciones detalladas, etc.) y suelen estar enfocados a la obtención de algún tipo de diploma o certificación. Por tanto requieren un esfuerzo constante y duradero.

Otra manera de enfocar la enseñanza, sobre todo cuando quieres ampliar algo sobre la que ya tienes una base, es a base de ejercicios rápidos. Durante bastantes años se han usado los *jueces online* o *correctores automáticos*. Hay un gran análisis sobre este tema y como poder evolucionarlo en el informe titulado 'Estudio de viabilidad de un entorno de aprendizaje colaborativo de lenguajes de programación'[13], realizado por profesores de esta misma universidad. En el se explica como normalmente este tipo de sistemas, aplicados a lenguajes de programación, se basan en la ejecución de unos casos de prueba. Esto exige que los instructores desarrollen previamente estas pruebas, tarea muchas veces poco grata.

Saliendonos del mundo de la programación, hay otros ejemplos, como el de Duolingo[5] que permiten aprender un idioma a partir de otro que ya sabes previamente. Sus características son muy interesantes, y destacan:

- Plantear el aprendizaje como un juego haciendo que el usuarios gane puntos y experiencia según va avanzando. Además incluye vidas, que hacen centrar la atención en la tarea presente, para no tener que reiniciar el nivel.
- Guarda información sobre los fallos del usuario, para intentar repetir esas preguntas y que aprenda los concepto de manera definitiva.
- Incluye prácticas con tiempo, y la posibilidad de certificar el nivel del idioma mediante tests online.

Hay otras alternativas como Bussu[4], donde los usuarios hacen simultáneamente de alumnos y profesores e interactúan entre ellos en una red social con el objetivo de aprender otro idioma.

3.3. Influencias tecnológicas

Antes de definir como llevar a la práctica este proyecto quisimos ver desde alto nivel, que tipo de sistema queríamos desarrollar. Durante la carrera hemos aprendido entre otras cosas lenguajes de programación que nos permiten hacer software ejecutable en ordenadores de sobremesa, en asignaturas como FP[6], o TP[11]. Aunque este podría haber sido un enfoque válido, no era el que queríamos seguir. Nos parece que podríamos hacer algo mucho más rápido de probar y usar por primera vez, sin la necesidad de instalar pesado software adicional.

En otras asignaturas como SC[9] aprendimos a montar y configurar un CMS (Content Management System, es decir sistemas de gestión de contenidos) y con ello montamos una Web. Esta encuadre nos gusta más, pues permite a cualquiera que tenga un navegador de Internet acceder a nuestra plataforma. Sin embargo los CMS actuales no permiten hacer cosas tan concretas y específicas como lo que queríamos hacer. Además, dejaríamos sin usar la mayoría de las características de estos, y pensamos que no usaríamos todo el potencial que ofrecen estas plataformas.

También cursamos AW[2], donde aprendíamos a desarrollar una Web desde el principio. Realizar un desarrollo de este tipo nos parecía sumamente interesante, pues nos permitiría un alto grado de flexibilidad con la manera en que queremos realizar el proyecto, y permitiría a los potenciales usuarios probar y usar la plataforma de manera rápida. En esta asignatura también nos apoyábamos en conocimientos adquiridos en otras como BD[3] y ABD[1], donde aprendimos a montar bases de datos y a consumirlas desde las aplicaciones. Con todos estos conocimientos podemos montar un sitio Web interactivo y que guarde una gran variedad de datos.

También sentíamos particular inclinación por hacer que el contenido estuviera disponible en dispositivos móviles, bien sea adaptando la Web o a través de una aplicación móvil. Algunos estábamos matriculados en el curso de realización de la asignatura en PAD[8], y vimos una oportunidad para poner de manifiesto lo que se podría aprender en esa asignatura.

3.4. Propuestas y objetivos

Gracias a la investigación de todo lo expuesto anteriormente, fijamos nuestras ideas. Nos decidimos a hacer un *front-end* para una herramienta que se iba a basar en el informe anteriormente citado[13]. Esto es, hacer un proyecto colaborativo para aprender lenguajes de programación, similar a Duolingo[5]. Como iba a estar centrado en código de programación, decidimos llamarlo DuoCode, pues aprenderás lenguajes de programación nuevos a partir de otros que ya sabes.

Para realizar este sistema, pensamos que sería importante separar la parte puramente Web de la que se conecta con la base de datos. De esta manera la aplicación será mucho más modular, y podría ser realizadas aplicaciones móviles que usen los mismos datos. Pensamos en realizar una Web con una interfaz limpia, y a ser posible que sea *Responsive*, es decir que tenga un diseño adaptable a distintos dispositivos. Las principales características que queríamos que tuviese disponibles el usuario son:

- Poder seleccionar el idioma de programación que sabes y el que quieres aprender. El sistema te mostrará código en el lenguaje que dominas y te pedirá rellenarlo en el que no.
- Habrá una clara organización, con código agrupado por temas.
- El usuario tendrá una puntuación que irá aumentando según avance. Además los ejercicios estarán agrupados, y tendrá una serie de vidas para completarlos. Esto favorecerá que el usuario se fidelice con la aplicación.
- Cuando el usuario falle un ejercicio pero no esté de acuerdo, podrá proponerlo como candidato a válido. Con la ayuda de la comunidad y de moderadores, estos se podrán incorporar como soluciones en un lenguaje de programación determinado.
- Se podrá guardar ejercicios favoritos para poder consultarlos de nuevo.
- Habrá una interacción con las redes sociales. Se podrá hacer login con alguna de ellas, y compartir tus hazañas en esta.

Referencias

- [1] Ampliación de base de datos.
- [2] Aplicaciones web.
- [3] Bases de datos.
- [4] Bussu. <https://www.busuu.com/es/>.
- [5] Duolingo. <https://es.duolingo.com/>.
- [6] Fundamentos de la programación.
- [7] Moodle. <https://moodle.org/>.
- [8] Programación de aplicaciones para dispositivos móviles.
- [9] Software corporativo.
- [10] Stack overflow. <http://stackoverflow.com/>.
- [11] Tecnologías de la programación.
- [12] Wikipedia. <https://es.wikipedia.org>.
- [13] A. R. (coordinador). Estudio de viabilidad de un entorno de aprendizaje colaborativo de lenguajes de programación. Informe del PIMCD 97/2014, 2015.

A. Primera sección del apéndice