

Tándem

Alberto López Gutiérrez

Resum—La finalidad de Tándem es poder realizar un modelo de negocio centrado en ofrecer una solución a diferentes usuarios que quieran aprender idiomas vía online en cualquier parte del mundo. El objetivo es crear una plataforma online totalmente gratuita de cara a los usuarios combinándola con una parte de negocio que incluiría academias de idiomas a las cuales les ofreceríamos la posibilidad de impartir clases online por medio de nuestra aplicación.

Paraules clau— Tándem, idiomas, academias, plataforma, aplicación, modelo de negocio, clases online, Hangouts, Front-end, Back-end, base de datos, MVC, PHPMYADMIN, CodeIgniter.

Abstract— The purpose of Tándem is to provide a solution to users who want to learn different languages online anywhere in the world. The aim is to create a completely free online platform for anyone worldwide. The business includes language schools where you will have the possibility of getting online classes through our application .

Index Terms— Tándem, languages, language schools, website, application, business model, online classes, Hangouts, Front-end, Back-end, database

- Las partes marcadas en gris son las partes comunes de los 3 miembros del grupo.
- Las partes propias están expresadas en primera persona.



1 INTRODUCCIÓN

A ctualmente, mucha gente tiene interés en aprender idiomas y una de las herramientas más populares para ello es Internet, ya que permite al usuario poder aprender un idioma con comodidad y sin la necesidad de desplazamiento. Además, es totalmente compatible con el horario del usuario, ya que se adapta a las necesidades de cada uno.

Nuestro proyecto quiere ser un punto de conexión entre diferentes personas que quieran realizar un retroaprendizaje entre ellos, siendo ellos mismos quienes enseñen y aprendan un idioma por medio de videoconferencia.

Otra ventaja que queremos ofrecer es la facilidad de que academias puedan ofrecer clases online a usuarios con diferente disponibilidad a partir de una cartera propia de profesores.

Por lo tanto, vamos a dividir el proyecto en 2 partes claramente diferenciadas: la parte de negocio y la parte técnica.

-
- E-mail de contacte: Alberto.lopez @e-campus.uab.cat
 - Menció realitzada: Tencologies de l'informació.
 - Treball tutoritzat per: Juanjo Villanueva (CVC)
 - Curs 2015/16

2 ESTADO DEL ARTE

Debido al crecimiento en auge del aprendizaje de idiomas y a la falta de tiempo de las personas para acudir a clases presenciales, nace una idea de proyecto. Esta idea de proyecto pretende desarrollar y mejorar las herramientas y aplicaciones web que hay actualmente en el mercado.

Encontramos plataformas tan asentadas como podrían ser Italki y Verbling, que cuentan con un gran número de usuarios y son mundialmente conocidas. Estas plataformas te permiten tanto contratar a un profesor como intercambiar idiomas con otros usuarios.

La idea de Tándem es utilizar los puntos fuertes y ventajas de estas herramientas ya asentadas y, a la vez, encontrar nuevos puntos para explotar y mejorar dichas herramientas

3 OBJETIVOS

Tándem nace con una propuesta inicial cuya base estaba en ofrecer la posibilidad de que varios usuarios se conectaran e intercambiaron diferentes idiomas mediante videoconferencias.

La idea principal era que los usuarios pudieran aprender idiomas de una forma amena, gratuita y online.

Para ello, estos usuarios se podrían conectar en diferentes salas, las cuales tendrían diferentes temas de interés. Di-

chos temas podrían oscilar entre categorías como “Deportes” o “Gastronomía”, etc. Para filtrar estas salas, el usuario tendrá que seleccionar uno o varios idiomas a aprender y uno o varios idiomas a enseñar. De esta forma, un usuario podría encontrar salas con los idiomas y temas de interés que desee. También, se querían preparar salas de estudio y de preparación de diferentes exámenes y pruebas, donde, por ejemplo, un usuario podría encontrar una sala con uno o varios usuarios preparándose para el examen del “First”.

3.1 Objetivos parte técnica

Queremos ofrecer una plataforma web donde los usuarios se puedan registrar y “logear” con su cuenta, pudiendo tener un listado de amigos/conocidos.

La funcionalidad básica ha de ser que un usuario pueda elegir entre crear una sala de chat o buscar una sala ya creada por otro usuario. Estas salas dispondrán de un número de usuarios (limitado en el momento de creación de la sala), uno o varios idiomas a enseñar, uno o varios idiomas a aprender y una temática. De esta forma, un usuario al crear/buscar una sala podrá elegir el idioma que domina y el idioma que desea aprender junto con una temática de su interés que haga más atrayente el aprendizaje.

Una vez haya un grupo de usuarios en una sala, mediante un sistema de videoconferencias podrán estar todos unidos en una misma llamada y gestionar ellos mismos cuánto tiempo quieren alargar la sesión y cuándo quiere cada uno abandonarla.

Como último punto a destacar, crearemos una sección de academias, dónde cada academia podrá generar su propia “landing page” y publicitarse a sí misma. Además, estas academias podrán ampliar su enseñanza presencial mediante clases online utilizando nuestra plataforma. Este último punto está directamente relacionado con el modelo de negocio, que veremos a continuación.

Desarrollo Back-end/Backoffice: Este apartado se puede considerar el que más peso tiene en mi parte de proyecto. Desarrollado con Codeigniter y utilizando el paradigma MVC.

Muy unido al Back-end nos encontramos con el Backoffice, éste nos permite la modificación de datos de la web sin tener que modificar código, con simples formularios para conseguir una buena usabilidad de la aplicación.

Seguridad: Desde un primer momento el apartado de seguridad se valoró con mucho peso, como se le quiere dar continuidad al proyecto/negocio y este va a ser accesible al público puede sufrir ataques de varios tipos, en un primer momento los ataques al servidor se van a frenar utilizando servicios cloud como Azure, Google o Amazon, sin embargo los ataques de software deben ser controlados totalmente desde nuestra aplicación, para ello nos encontramos con que CodeIgniter proporciona ciertas

librerías para controlar ataques de Sql Injection, Modificación de cabeceras, Cross-Site Scripting...

Controlador del MVC: Para la unión del modelo con la vista y darle lógica a la aplicación es necesario el desarrollo de los controladores del MVC, el controlador proporciona todas las funcionalidades de la web, alimentándose del modelo y proporcionando datos a la vista.

Cloud Services: Nos encontramos con dos necesidades que nos hicieron plantear la contratación de Cloud Services, la seguridad de la parte del servidor y la posible demanda de computo mientras va creciendo el uso de la web, a la vez tenemos la ventaja de que el pago por los servicios va en relación al uso de la web, por lo que no será necesario un gasto muy elevado en la fase de lanzamiento.

Feedback de clientes: Una parte importante a la hora de proceder a avanzar con una idea de negocio es obtener la opinión de los diferentes usuarios y clientes que van a utilizar la plataforma, con los usuarios hemos pasado encuestas y con las academias hemos realizado varias reuniones.

3.2 Objetivos parte negocio

La parte de negocio ha sido muy importante para nosotros des de un principio, ya que este proyecto va enfocado a un futuro negocio. Así que nuestra pregunta era: ¿cómo podemos hacer para que este proyecto se convierta en un negocio pero que a la vez nos proporcione los suficientes usuarios para que la propia plataforma se sostenga en sí misma? Es decir, necesitamos un buen número de usuarios para que la web tenga una fluidez constante de actividad pero también buscar una fuente de ingresos para convertirlo en un negocio.

Básicamente, la idea es dividir el proyecto en 2 partes:

La primera sería un tandem para que personas de todo el mundo intercambien idiomas, pero al final convirtiéndose en una red social de alumnos y profesores, ya que cada usuario tiene la posibilidad de convertirse en un miembro vip/profesor y poder establecer un contrato con nosotros. Para llegar a convertirse en un profesor, cada usuario recibirá un feedback al terminar una reunión por parte del resto de integrantes, dónde se evaluará su dominio de la lengua así como la habilidad y soltura para enseñar. Una vez un usuario reciba muchos (posiblemente cientos) feedbacks positivos, se le dará la oportunidad de formar parte de nuestra plantilla de profesores, los cuales tendrán la opción de recibir una remuneración por impartir clases en la plataforma. Tandem se llevaría una pequeña parte de estos pagos. Con esta medida, queremos fomentar y ampliar el número de usuarios en nuestra web ya que tendrán la posibilidad de llegar a ganar dinero incluso sin tener un título como tal, a la vez que están aprendiendo otros idiomas.

La segunda parte implica a las academias. Queremos ofrecerle a las academias una cartera de profesores para que puedan utilizarlos, bien sea para que los alumnos practiquen el idioma con gente nativa de otros países, como para utilizar enseñanza online y realizar horas extraescolares. Las academias nos pagaría por nuestra cartera de profesores y también por publicitarlas en nuestra plataforma. Además, las academias dispondrían de un espacio propio en nuestra plataforma donde podrán ellas mismas gestionar todas sus opciones. Esta medida sería la parte más importante del modelo de negocio.

Además, dispondrías de un pequeño espacio para introducir pequeños spots publicitarios relacionados con la temática de la web, como por ejemplo mediante la publicidad de Google. En un futuro, se valoraría dar la posibilidad a las academias a poner sus propios vídeos publicitarios en este espacio para que puedan salir en el resto de páginas, no solo en su propia "landing page".

En la parte de anexos podremos encontrar el canvas del modelo de negocio.

4 METODOLOGÍA

Nuestra metodología se basa en reuniones constantes con el tutor Juanjo Villanueva. En estas reuniones comentamos el proceso actual en el que se encuentra el proyecto, le enseñamos el trabajo realizado semanalmente, aclaramos diferentes dudas y definimos objetivos para la semana siguiente. Realmente nos está siendo muy útil esta metodología de reuniones semanales con él ya que es mucho mas fluida la comunicación y nos ayuda a poder marcar objetivos semanales.

La metodología de trabajo que estamos realizando no ha cambiado. Seguimos utilizando claramente la metodología scrum, la cual nos está siendo muy efectiva. Hemos podido llegar claramente a la mayoría de las tareas propuestas por "sprint" y nuestra coordinación ha mejorado enteros, claramente influenciada por el rodaje y entendimiento del grupo que va en auge a medida que las semanas transcurren.

Como punto negativo, comentar que con Juanjo no hemos podido hacer todas las reuniones que nos hubiera gustado ya que hemos ido muy cargados tanto laboral como académicamente y hemos tenido que aplazar algunas reuniones. La tónica ha sido hacerlas como máximo cada dos semanas en vez de semanalmente. Igualmente las reuniones con él han sido muy provechosas y siempre nos ha conseguido centrar en los objetivos.

5 DESARROLLO DEL BACKEND/BACKOFFICE

5.1 Primer análisis

Después de varias reuniones con el equipo, tutor y después de los primeros bocetos del Front-end, recoger funcionalidades y tener claro que la usabilidad era una parte muy fuerte en la web, tuve siempre en mente que el Backoffice era primordial para cumplir todos los objetivos.

Junto con el Back-end de los usuarios en la gestión de las salas, perfil, mensajería privada, gestión de amigos, etc. También era necesario un Backoffice para administradores donde poder recoger estadísticas de la web tales como uso, tiempos de carga, origen de las visitas, etc. y poder administrar usuarios, dar de alta a academias, introducir nuevas categorías, imágenes preestablecidas, idiomas y academias.

5.1 Codelgniter

5.1.1 Framework

Para el desarrollo del Back-end elegí utilizar un Framework en vez de desarrollarlo directamente sobre un lenguaje de programación como podría ser PHP, Java o .NET.

Un Framework está basado en un lenguaje de programación, pero este aporta funciones, estándares y facilidades para la programación añadidas al lenguaje, así como una sencilla reutilización del código. Más profundamente entenderíamos un Framework como "un lenguaje de programación" que utiliza otro lenguaje por debajo y traduce las sentencias introducidas en este mismo en el lenguaje de programación en el que está basado.

5.1.2 La elección de Codelgniter

Desde el primer momento en el que se decidió que me encargaría de la parte del Back-end tuve claro que utilizaría o bien PHP, o bien un Framework basado en este, debido a que el lenguaje de programación de parte de servidor con el que me siento más cómodo y he utilizado más ha sido PHP.

Finalmente y debido a que nunca había utilizado un Framework y tenía ganas de aprender a programar en uno, decidí que utilizaríamos CodeIgniter, la instalación del mismo es sencilla:

- Descargar y descomprimir el Framework desde su web oficial.
- Instalar un servidor web, en nuestro caso un Apache a través del paquete de WAMP.
- Configurar un virtual host para el proyecto.

- Añadir los archivos descomprimidos al directorio htdocs de apache (Figura 1).

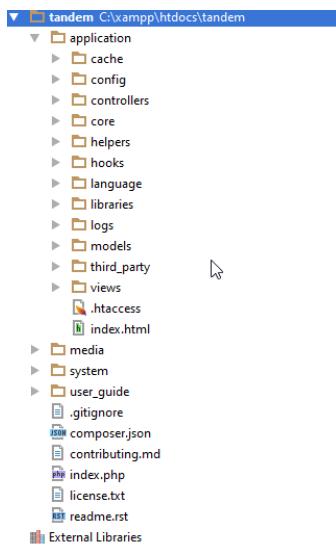


Figura 1

Una vez instalado CodeIgniter falta configurarlo, para ello hay que modificar diferentes archivos de CodeIgniter para realizar la conexión con la base de datos, modificar el controlador principal, habilitar URLs, etc...

5.1.3 CodeIgniter vs PHP

Me gustaría destacar algunas de las funcionalidades que ofrece CodeIgniter respecto a PHP:

Creación de Sesión:

A continuación se encuentran las diferencias entre crear una sesión en CodeIgniter(Figura 2) y PHP(Figura 3) como se puede observar en CodeIgniter con una línea se crea la sesión sin embargo en PHP necesitamos de al menos 3 líneas.

```
$this->session->set_userdata($data);
```

Figura 2

```
$_SESSION['username'] = $_POST['username'];

if(empty($_SESSION['username'])) {
    header('Location: index.html');
}
```

Figura 3

Carga de modelos en CodeIgniter vs Include de archivos en PHP:

Como podemos observar la manera de cargar vistas desde CodeIgniter (Figura 4) es mucho más intuitiva al llamarla desde el controlador que la manera de incluir archivos completos desde PHP desde la propia vista(Figura

5).

```
$this->load->view('index');
$this->load->view('login');
```

Figura 4

```
include('login.php');
include('header.php');
```

Figura 5

5.2 Back-End

Como hemos comentado anteriormente una parte muy importante del Back-end es agilizar la creación de salas, modificación del perfil, etc. Utilizando Jquery para los campos de los formularios y generando listas autocompletadas a través de CodeIgniter.

A continuación muestro unos ejemplos sencillos para la creación del Back-end.

En la Figura 6 podemos observar un ejemplo para el listado de las categorías en la creación de las salas.

```
<?php
$first = false;
foreach($categories as $row){
    if(!$first) {
        echo "<option selected>---Categoria---</option>";
        $first = true;
    }
    echo "<option>".$row->name_catRoom."</option>";
}
?>
```

Figura 6

En la Figura 7 muestro como estamos haciendo el Login de los usuarios, en primer lugar se comprueba si el usuario existe, en caso afirmativo se genera una Array donde se guardan los datos que más se van a usar del usuario, a continuación esos datos se guardan como variable de sesión.

```
$user = $this->modeluser->login($username,$password);

if($user == TRUE)
{
    $data = array(
        'is_logged_in' =>      TRUE,
        'mail'           =>      $user->mail,
        'id_user'        =>      $user->id_user,
        'username'       =>      $user->username,
        'id_user'        =>      $user->id_user,
        'avatar'         =>      $user->avatar,
    );
}

$this->session->set_userdata($data);
```

Figura 7

5.3 Backoffice

El Backoffice para los administradores está preparado para incrustar el Script de Google Analytics para obtener estadísticas de visitas tales como procedencia, nacionalidad, así como modificar las tablas de la Base de Datos.

6 SEGURIDAD

6.1 Primer análisis

Desde un primer momento y desde la primera reunión tuve claro que la Seguridad en nuestra Web era prioritaria, si cumplimos con las metas propuestas y conseguimos contar con un buen número de usuarios y academias, nuestra web va a ser objetivo de ataques. Aunque en un primer momento se puede pensar que webs de este contenido no suelen ser un objetivo de grupos de Hackers, sí que lo son por parte de usuarios independientes.

Desde CodeIgniter hemos incorporado seguridad para prevenir de los siguientes ataques:

6.2 SQL-Injection

Con los ataques de inyección SQL se pretende comprobar si las consultas que se realizan desde la web a la base de datos son robustas y seguras, por lo que si estas no están correctamente programadas se podría autenticarse en la web sin necesidad de conocer contraseñas o incluso realizar un listado de toda la base de datos.

En la Figura 8 se puede observar una consulta a la base de datos, para prevenir ataques SQL-Injection se utiliza el método “escape” del modelo DB para que omita caracteres que pueden dar error como, las comillas simples o dobles, barras, etc...

```
$email= $this->input->post('email');
$query = "SELECT * FROM subscribers_tbl WHERE user_name='".$this->db->escape($email);
$this->db->query($query);
```

Figura 8

6.3 Cross-Site Scripting (XSS)

Los ataques de Cross-Site Scripting consisten en modificar valores que la aplicación web usa para pasar variables entre dos páginas. Un clásico ejemplo de esto es hacer que a través de un buscador se ejecute un mensaje de alerta en JavaScript. El atacante podría robar las cookies para luego robar la identidad.

En la Figura 9 se observa una manera de como con CodeIgniter se pueden prevenir este tipo de ataques.

```
$data = $this->security->xss_clean($data);
```

Figura 9

El filtro de prevención `xss_clean` que ofrece CodeIgniter nos permite evitar las técnicas más comunes para el robo de cookies, aparte de esta técnica existen otras para tratar imágenes o ataques de CSRF.

7 CONTROLADOR DEL MVC

7.1 MVC

Para entender cómo funciona un controlador en CodeIgniter debemos entender también como funciona el paradigma MVC, MVC(Figura 10) fue creado para obtener una estructura de código y directorios limpia y ordenada, su finalidad es separar la parte de datos, lógica y vista de la aplicación.

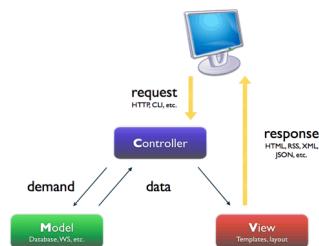


Figura 10

Con una rápida definición podemos definir que el modelo contiene todos los métodos o funciones necesarias para acceder a los datos, la vista contiene toda la parte visual de la aplicación y el Controlador hace de unión entre ambos, pide datos al modelo, carga vistas, entrega datos a la vista, recibe las peticiones de los usuarios, se encarga de enrutar a otros controladores o vistas, etc...

7.2 Como es un controlador en CodeIgniter

CodeIgniter está basado en el paradigma MVC pero añadiendo ciertos elementos, en la Figura 11 se puede observar como es la lógica interna que utiliza para servir un recurso.

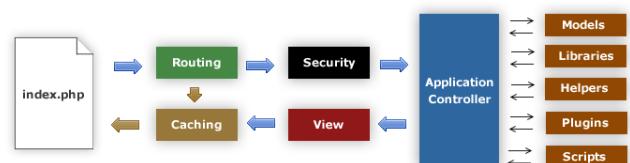


Figura 11

- Index.php es el controlador principal de la aplicación, inicializa los recursos para poder utilizar CodeIgniter.
- Una vez Index.php recibe una petición esta se envía al “Router” de CodeIgniter el cual decide en primer lugar si se enruta y en caso afirmativo cómo será la ruta.
- Antes de que el Controlador sea cargado, la petición se deriva al filtro de seguridad de CodeIgniter donde se examina cualquier dato que el usuario a enviado y la petición HTTP.
- Se carga el controlador y este a su vez carga los modelos, plugins o Scripts necesarios para completar la petición.
- El Controlador carga una vista enviándole los datos necesarios para su renderizado.

7.3 Análisis de los Controladores

Habitualmente cada Vista del Front-End lleva ligado un Controlador y cada controlador utiliza uno o varios Modelos de la aplicación, para poder desarrollarlos correctamente sin tener las vistas del Front-End o las funciones del Modelo fue necesario marcar desde un principio todas las funcionalidades de la Web en cada Vista, marcar nombres en común de las funciones, de los atributos, de las arrays de datos, etc...

Se necesitaba un gran número de controladores y tener todo organizado y localizable, con nombres mnemotécnicos era vital para poder encontrar rápido las dependencias entre controladores, modelos y vistas.

7.4 Implementación de un Controlador

Como hemos dicho anteriormente lo primero que hay que hacer en un Controlador es cargar los Plugins y Modelos(Figura 12), a continuación realizar las funciones necesarias(Figura 13) y por último cargar las vistas enviando datos(Figura 14).

```
public function __construct()
{
    parent::__construct();
    $this->load->model('modelCountryLanguage');
    $this->load->model('modelRoom');
    $this->load->library(array('session', 'form_validation'));
    $this->load->helper(array('url', 'form'));
```

FIGURA 12

```
if($this->isRoomCreated($this->session->userdata('id_user'))){
    $this->load->view('myRooms', array('languages' => $this->getLanguageList(), 'categories' => $this->getCategoryList(),
    'boolRoom' => $this->isRoomCreated($this->session->userdata('id_user')), 'room' => $this->getRoomInfo($this->session->userdata('id_user'))));
}
else{
    $this->load->view('myRooms', array('languages' => $this->getLanguageList(), 'categories' => $this->getCategoryList(),
    'boolRoom' => $this->isRoomCreated($this->session->userdata('id_user'))));}
```

FIGURA 13

```
$this->load->view('header');
$this->load->view('fringe');
$this->load->view('login');
```

FIGURA 14

8 CLOUD SERVICES

Para el hosting de la web se decidió que contratar servicios cloud, consideramos que es la mejor solución tanto en presente como en futuro.

Para empezar y muy ligado a la seguridad tenemos que de la administración del servidor, uso de Firewalls, etc, está en mano del proveedor y por otro lado nos permiten realizar pago por uso y una fácil ampliación de la capacidad de computo según las necesidades.

De entre las tres opciones analizadas, Amazon AWS, Microsoft Azure y Google Cloud Platform, hemos decidido

utilizar Google Cloud Platform.

8.1 GOOGLE CLOUD PLATFORM

Se ha decidido utilizar Google Cloud Platform principalmente porque vamos a utilizar la API de Hangouts y el Script de Google Analytics, en un futuro también es posible que utilicemos Google Apps for Work para el mailing.

Tener todos los servicios contratados con la misma compañía nos proporcionara facilidades para la administración, así como una mejor integración de todos los servicios entre ellos.

9 FEEDBACK DE CLIENTES

Durante el transcurso del proyecto he ido haciendo encuestas a los usuarios y entrevistas con las academias, junto con el equipo y tutor hemos pensado en las preguntas necesarias para acabar de definir el modelo de negocio y poner énfasis en algunas características.

Las encuestas a los usuarios las he realizado a través de la plataforma SurveyMonkey, web especializada en encuestas, las entrevistas con las academias constaban de una explicación del portal y una Batería de entre 10 y 15 preguntas de desarrollar, la duración estaba entre los 15~25 minutos.

10 RESULTADOS

Considero que los resultados obtenidos han sido totalmente positivos, hemos tenido problemas en el desarrollo como veremos más adelante, pero considero que hemos sacado el proyecto adelante y hemos podido llegar a un mínimo producto viable.

A continuación mostraré los resultados de los puntos trabajados y definidos anteriormente.

10.1 CodeIgniter

10.1.1 Opinión sobre CodeIgniter

Finalmente la elección de CodeIgniter para hacer el Back-End ha sido un acierto absoluto, aunque al inicio del desarrollo tuve que invertir bastante tiempo en aprender a usarlo, después de superar la curva de aprendizaje, puedo decir que programar era más rápido que con PHP, más ordenado y con más seguridad.

Algunas de las ventajas que he podido extraer como conclusión al utilizarlo han sido:

- Uso de MVC, el Framework obliga a utilizar el patrón MVC para el desarrollo de la aplicación, gracias a ello obtenemos una estructura limpia y

entendible para diferentes programadores que necesiten acceder al código.

- Librerías, gracias a las librerías podemos reutilizar código externo o implementar funcionalidades. CodeIgniter de por si trae añadida una librería para evitar ataques de tipo SQL-Injection.
- Reducción del tiempo de desarrollo, una vez superada la fase de aprendizaje, que en según que Frameworks suele tener una curva pronunciada, nos encontramos que programar en el Framework suele ser mucho más ágil que programar en el lenguaje en el que está basado.

Como desventajas podría destacar las siguientes:

- Curva de aprendizaje, en mi caso era la primera vez que utilizaba un Framework, conocía el lenguaje de programación en el que estaba basado, pero aun así me ha costado al menos un 10% del tiempo de desarrollo del proyecto poder aprenderlo para ir más rápido que programando en el lenguaje en el que está basado.
- Menos comunidad, si programamos por ejemplo en PHP podemos observar que hay una gran comunidad a su alrededor y la mayoría de dudas que puedan surgir están resueltas, sin embargo cuanto más específico sea un lenguaje(en nuestro caso un Framework) menos comunidad tendremos disponible.

10.1.2 Back-end, Backoffice y CMS

En lo que respecta al Back-end voy a mostrar un par de ejemplos de uso en la propia web.

El primer ejemplo es la impresión de los idiomas al crear la sala (Figura 15)

```
foreach($languages as $row){
    if(!$first){
        echo "<option selected>---Idioma a aprender---</option>";
        $first=true;
    }
    echo "<option>".$row->name_language."</option>";
```

Figura 15

El segundo ejemplo que quiero mostrar es como cambiamos los iconos de capacidad/ocupados de la sala(Figura 16)



Figura 16

El siguiente código en CodeIgniter se encarga de ello (Figura 17)

```
<?php $iter = 0;
$red=element('capacity', $room)->element('connected', $room);
while($iter < $red){
    echo "<img src='".$base_url()."media/img/rooms/conectado.png' width='20px' height='20px'>";
    $iter++;
}
$iter=0;
while($iter < $red){
    echo "<img src='".$base_url()."media/img/rooms/no-conectado.png' width='20px' height='20px'>";
    $iter++;
}
?>
```

Figura 17

Para gestionar fácilmente los datos que se muestran en la página, añadir categorías, cambiar títulos, gestión de salas, etc... Fue necesario hacer un gestor de contenidos(CMS en adelante), con el CMS pretendemos agilizar la introducción, modificación y eliminación de esos datos.

El primer ejemplo que quiero mostrar es la gestión de salas (Figura 18), como se puede observar desde ese menú del CMS se pueden eliminar salas y añadir salas.

ID	Sala	Usuario	Action
1	Speak with me	Urek	
2	Football talk	Messi	
3	Music	Bias	
4	General	Admin	
5	Castellano	francisco	
6	Aleman	Test	

Figura 18

A continuación muestro el formulario para añadir sala(Figura 19)

ID	Categoría	Action
1	Deportes	
2	Products	
3	Blogs	

Figura 19

Por último muestro el menú para gestionar categorías(Figura 20)

Título	<input type="text"/>
Enseño	<input type="text"/>
Aprendo	<input type="text"/>
Categoría	<input type="text"/>
	<input type="button" value="Guardar"/>

Figura 20

10.2 Seguridad

Tal como hemos visto, CodeIgniter, incorpora librerías para la seguridad de la web(desde el punto de vista de software), estas librerías se encuentran implementadas en todas las consultas e intercambio de datos entre controladores y vista. Para poder comprobarlo hicimos varios tests manuales, en ningún test de SQL apareció ningún error de la BBDD ni se dio ninguna información del sistema.

El robo de cookies en un principio no es posible debido a que no utilizamos cookies de momento, lo hacemos todo por sesiones, aunque están implementados los mecanismos para evitarlo.

10.3 El uso de los controladores

Anteriormente he mostrado como es un controlador de CodeIgniter, en este punto voy a mostrar cómo han quedado los controladores (Figura 21) y un ejemplo práctico de como lo hemos implementado.

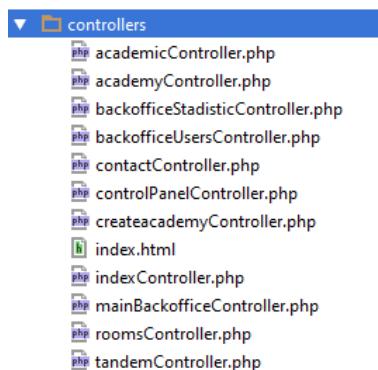


Figura 21

Para mostrar el ejemplo de cómo funcionan las rutas y cambios de vista en CodeIgniter, voy a proponer el ejemplo práctico de ir desde la página principal de Tándem (una página informativa) a la página de inicio (Tándem) de la aplicación (Solo se puede acceder estando logeado)

En primer lugar indicamos cual es el controlador por defecto que se va a cargar al acceder a la Web desde el archivo routes.php (Figura 22).

```
$route['default_controller'] = 'indexController';
```

Figura 22

Una vez se accede al indexController se comprueba si el usuario esta logeado, para dirigirlo al controller de Tándem o para mostrar la vista de index y login (Figura 23).

```
public function index()
{
    if($this->session->userdata('is_logged_in')){
        redirect(base_url('index.php/tandemController'));
    }else{
        $this->load->view('index');

        $this->load->view('login');
    }
}
```

Figura 23

Si suponemos que el usuario no está logeado se cargarán las vistas index y login, en caso de que lo este se le redirige al controlador tandemController, en ese controlador se realizará una comprobación similar para decidir si cargar las vistas o hacer una redirección.

10.4 Google Cloud Platform

Google Cloud Platform ofrece diferentes modalidades de servicios, pruebas y siempre es posible utilizar todo el ecosistema creado por Google (APIs, Apps for Work...)

Quiero destacar de entre toda la información recopilada la siguiente:

- Al registrarnos en la plataforma se obtienen 300\$ para gastar en Cloud Platform durante 60 días.
- El periodo de prueba de 60 no obliga a ninguna permanencia.
- Módulo de App Engine que permite escalar los recursos disponibles automáticamente de manera dinámica disponiendo de una disponibilidad del 99.95%
- Google Cloud SQL: El sistema de Google se encarga automáticamente de truncar Logs, aplicar parches de seguridad o replicar las BBDD.
- Precio inicial según nuestras necesidades en el lanzamiento menor de 30€/mes.

10.5 Feedback

A continuación voy a exponer algunas de las conclusiones que hemos sacado en las entrevistas con las academias.

Por lo que respecta a las academias estas se sienten atraídas por la posibilidad de ofrecer más idiomas sin necesitar buscar un profesor, unido a que también tienen interés en poder realizar clases fuera del horario de la Academia

También nos comentan que es una idea que puede calar en grandes ciudades como Barcelona donde se busca

aprender más variedad de idiomas que en una ciudad más pequeña como podría ser Barbera del Valles (Ciudad de la academia entrevistada).

La academia "Whats Up" nos explicó que ellos no tienen en estos momentos página web y les gustaría un pack para poder crear una de manera sencilla, que no les importaría que fuera genérica pero si querían que fuera "bonita" y que pudieran poner todos los servicios que ofrecen.

11 PROBLEMAS

El principal problema que me he encontrado al realizar el TFG es la coordinación, aunque creo que lo hemos llevado muy bien es cierto que muchas veces nos ha costado poder coordinarnos correctamente, o alcanzar ciertos objetivos del Sprint para no entorpecer a los compañeros, también en ocasiones no hemos podido realizar la reunión semanal con Juanjo y hemos tenido que retrasarla una semana más.

Otro "problema" ha sido la curva de aprendizaje de CodeIgniter, aunque considero totalmente positivo el uso del Framework para desarrollar el Back-end, es cierto que al inicio del desarrollo me pregunto varias veces si había sido una buena idea. Después de las primeras semanas y al acabar los primeros controladores me di cuenta del potencial que tenía y de lo bien estructurado que me estaba quedando el código y directorios.

12 CONCLUSIONES

La primera conclusión que se me viene a la mente es el aprendizaje que me llevo, no solo de mi parte sino también de la parte de mis compañeros, hemos compartido cualquier nuevo conocimiento entre los integrantes del grupo y ha sido más fácil lidiar con el aprendizaje de las tecnologías utilizadas.

Respecto al proyecto creo que estamos en la línea buena, hemos visto, como llevamos viendo durante todos los estudios en la carrera, que cualquier funcionalidad que pueda parecer fácil acaba complicándose y no se puede dar por hecho nada hasta que está hecho.

Por otro lado ha sido muy positiva la tutorización de Juanjo, nos ha ayudado a entender los intríngulis de un negocio, a animarnos a pivotar y a ver los cambios necesarios en la aplicación si queríamos enfocarla como el eje de un negocio.

Por último me llevo ilusión, ilusión de seguir adelante con el proyecto, con mis compañeros y peleando por sacarlo, es posible que al final no llegue a ningún puerto y el barco se hunda en el océano, pero aun así está valiendo la pena intentarlo y creo que continuará valiéndola.

13 LÍNEAS FUTURAS

La primera meta de futuro que me quiero marcar antes de seguir adelante con la implementación de nuevas funcionalidades es acabar de pulir todos los métodos que tenemos actualmente, testear todos los casos posibles y dejar el código lo más pulido que se pueda.

Una vez pulido todo el código continuar con la implementación de más funcionalidades, tales como roles de usuarios, nuevas opciones para las academias, gestión automática de reservas, sistema de ticketing, etc... Finalmente hacer un Foro interno.

14 AGRADECIMIENTOS

Quiero agradecer a mis compañeros la implicación en el proyecto y el conocimiento que han aportado, ya que sin ellos esto no habría sido posible, agradecer a todos los profesores que he tenido en mi vida académica sus conocimientos aportados y animarles a que sigan haciendo su trabajo e impartiendo sus conocimientos con la misma fuerza que lo hicieron conmigo, agradecer a nuestro tutor Juanjo la implicación en el proyecto y la tutorización del mismo, así como sus consejos y ánimos, agradecer especialmente a toda mi familia el haber estado siempre ahí, animarme en los momentos de bajón y ayudarme a sacar adelante el sueño de sacarme una carrera, por último agradecer al lector que en estos momentos lee este párrafo el haberse interesado por la lectura del Artículo.

A todos gracias.

15 BIBLIOGRAFIA

- [1] Bootstrap [Online]. Available: <http://getbootstrap.com/>
- [2] CODEIGNITER manual [Online]. Available: http://www.codeigniter.com/user_guide/
- [3] HANGOUTS [Online]. Available: <https://developers.google.com/+hangouts/>
- [4] Canvas [Online] Available: <http://innokabi.com/canvas-de-modelo-de-negocio>
- [5] Start up [Online] Available : http://es.wikipedia.org/wiki/Compa%C3%B1a_startup

APÈNDICE

A1. CANVAS

Key Partners	Key Activities	Value Propositions	Costumer Relationship	Customer Segments
No existen	 Plataforma Videoconferencias. Foro. Cartera Profesores. Buscador de Academias. Variedad de Idiomas. Retroaprendizaje.	 Comodidad para aprender idiomas. Precio, enseñanza de calidad, usuarios nativos.	 Redes Sociales. Fidelización(usuarios VIP)ç	 Academias. Personas interesadas en aprender idiomas.
Key Resources	Channels			
	 Relaciones sociales, temas de interés. Servicio vide conferencia Foro. Blog.	 Publicidad en academias Cartera usuarios VIP Cartera Profesores	 Profesores. Escuelas	
		 Portal Web. Redes Sociales		
Cost structure	Revenue Streams			
Servidor web(Amazon/Azure) Dominio Personal		 Academias ofreciendoles una cartera de profesores + buscador Google AdWords Programa de afiliados. Sistema de cobro por clase impartidas mediante la aplicación		