

Tándem

Marc Barnús García

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 Tandem 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— Tandem, languages, language schools, website, application, business model, online classes, HANGOUTS, Front-end, Back-end, database, , MVC, PHPMYADMIN, CODEIGNITER.

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



1 INTRODUCCIÓN

Actualmente, 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 contacto: marc.barnus@e-campus.uab.cat
 - Mención realizada: Enginyeria del Software.
 - Trabajo tutorizado por: Juanjo Villanueva (CVC)
 - Curso 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. Dicho 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, donde 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 de la Base de datos: Mi objetivo principal referente a este proyecto era crear el núcleo de la base de datos, esta parte es muy importante ya que todos los datos que debe contener la aplicación son guardados en una base de datos para posteriormente recuperar correctamente estos datos en cada sección o para cada usuario al que corresponda, para la implementación he utilizado el PHPMYADMIN de wamp basado en sql.

Desarrollo Modelo: Dentro del MVC el modelo es la parte que comunica con la base de datos y envía la información a las vistas a partir del controlador. Estos modelos implementan la conexión a la base de datos y sus consultas respectivas, en esta parte he tenido que comunicarme constantemente con los demás miembros del equipo para saber que recuperar en cada sección.

Desarrollo sección Academias: Una de las partes más importantes es la parte de nuestro cliente principal, las academias, para que tengan una gestión correcta he desarrollado un método simple y intuitivo donde puedan modificar cómodamente sus datos y la plantilla en uso sin ningún tipo de complejidad.

Requisitos funcionales: Referente a la definición de las funciones que el sistema debe cumplir así mismo como el objetivo que desarrolla esa funcionalidad.

Requisitos no funcionales: Referente más a las características de funcionamiento como el rendimiento, disponibilidad, accesibilidad, etc.

3.2 Objetivos parte negocio

La parte de negocio ha sido muy importante para nosotros desde 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 tándem 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. Tándem 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 pagarían 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 más 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 REQUISITOS

Una parte importante de la que yo me he encargado en este proyecto son los requisitos funcionales y no funcionales que la aplicación debe cumplir:(en orden de mayor a menor importancia)

- Requisitos funcionales:
 - Poder cargar todas las sales para poder mostrarlas al usuario y poder realizar su conexión.
 - Que el usuario pueda crear su sala especificando las características adecuadas.
 - El usuario debe poder filtrar las salas según especificaciones como puede ser el idioma a

aprender o la capacidad de la sala.

- El usuario debe tener la opción de poder modificar su sala creada.
- El usuario solamente puede tener una sala creada.
- El usuario debe tener la opción de poder eliminar la sala.
- La aplicación requiere de hacer login para poder acceder.
- El usuario tiene la posibilidad de establecer relación de amistad con otros usuarios.
- Las academias deben poder tener su propia parte de administración donde se les permitan modificar sus datos y su template.
- Para que una academia pueda estar dada de alta en nuestra aplicación debemos facilitarle nosotros el usuario y password(que podrán cambiar) de entrada una vez aceptada.
- No funcionales:
 - La aplicación debe ofrecer un alto porcentaje(>80%) de disponibilidad.
 - Debe ser segura sobre todo en la parte de datos de academias.
 - Debe ser muy intuitiva y fácil de utilizar.
 - La aplicación debe utilizar el estilo de arquitectura MVC.
 - Debe estar implementada a partir de CO-DEIGNITER.

6 DISEÑO Y IMPLEMENTACIÓN DE LA BASE DE DATOS

6.1 Primer análisis

El primer paso fue definir el contenido que iba a tener la aplicación, este contenido fue especificado después de bastantes reuniones entre los miembros del equipo y el tutor y con diferentes academias para facilitarnos el rumbo que debía tomar la aplicación. Despues de analizar la información obtenida pude comenzar a hacerme una idea del contenido y las tablas que debía tener la base de datos para poder mostrar la información necesaria y la relación entre las diferentes secciones y apartados que iba a tener la aplicación.

6.2 Diseño del modelo E/R Usuarios-Room

Después de realizar el primer análisis comencé a implementar lo que sería el modelo E/R(figura 1) de la base de datos definiendo las tablas que debe contener y los atributos de cada tabla así como su relación con el resto de tablas de la base de datos. Decidí separarlos en dos partes diferentes a medida que avanzaba el proyecto para poder mantener el contacto con los demás miembros del grupo y definir la información que les haría falta para sus respectivos Front-end y Backend/Backoffice por lo cual empecé primero a realizar el modelo E/R sobre los usuarios y las salas de la aplicación.

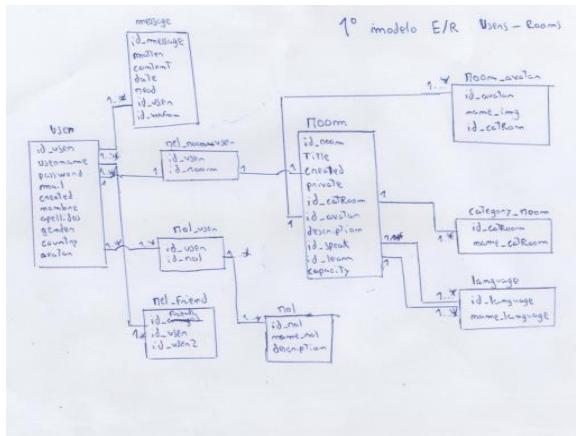


Figura 1

Específicamente esta parte contendría toda la información que puede realizar y/o ver un usuario que utilice la aplicación así como todo el contenido y características de las salas creadas por estos.

Definiendo un poco más las tablas: (Clave primaria en punto interior)

- **User:** Tabla que contiene toda la información referente a un usuario donde se guardaran sus datos, nombre, apellidos, correo, etc.
 - Mail
- **Rel_friend:** utilizada para crear la relación de amistad entre dos usuarios.
 - Rel_friend
- **Room:** Guardaremos la información que contendrá una sala creada por un usuario como el título, la capacidad, la descripción y diferentes claves externas que se relacionaran con diferentes tablas.
 - Id_room
- **Message:** Tabla que servirá de relación entre los mensajes que se envíen los usuarios.
 - Id_message
- **Languages:** contendrá un listado de todos los idiomas posibles para la relación de idioma a aprender y idioma a enseñar en la sala.
 - Id_language
- **Rol:** guardaremos los roles que puedan tener los usuarios, en un futuro se añadirán usuarios vip, usuarios Preparados para dar clases, etc.

- Id_rol
- **Rol_user:** tabla que relacionara cada rol con su usuarios.
 - Id_rol
- **Room_avatar:** donde se guardarán las imágenes que se mostrarán como parte de la sala.
 - Id_avatar
- **Room_category:** tendrá una lista de diferentes categorías que podrá tener una sala como deportes, gastronomía, cine, etc.
 - Id_catRoom
- **Rel_room_user:** Creara la relación entre el usuario y su sala creada.
 - Id_user
 - Id_room

6.3 Diseño del modelo E/R Academias

Una vez implementada la parte de usuarios-room ya pude proceder a la implementación de la parte de las Academias(Figura 2) referente al contenido y relaciones que debería tener su parte de base de datos para poder mostrar e modificar el contenido adecuado. Las academias no utilizarían un usuario normal de la tabla “user” para acceder sino que tienen una tabla aparte donde guardaríamos todos los usuarios que les proporcionamos nosotros mismos para su acceso.

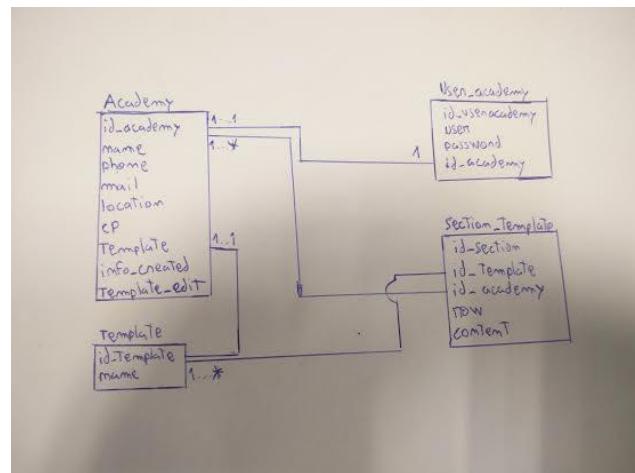


Figura 2

Definiendo un poco más las tablas: (Clave primaria en punto interior)

- **Academy:** Tabla que contiene los datos propios de una academia de información a mostrar cómo puede ser el teléfono, el código postal, etc.
 - Id_academy
- **Section_template:** guardaremos las secciones junto con su contenido de los templetes relacionados a su academia correspondiente.
 - Id_section
- **Template:** Tendrá los nombres de los diferentes templates que puedan escoger las academias para mostrar su información y modificarla.

- Id_template
- **User_academy:** Esta tabla tiene los usuarios que utilizaran las academias para poder acceder a su parte de administración, este usuario primamente será facilitado a una academia por nosotros o un administrador de tal forma que conste que tiene acceso al sistema de la aplicación, luego proporcionarle la opción de que ellos mismos puedan cambiar la password por el tema de protección de datos.
- Id_useracademy

6.4 Implementación con PHPMYADMIN

Ya realizados y comprobados los modelos E/R comencé la implementación en PHPMYADMIN que es una herramienta contenida en wamp/xampp para la implementación de la base de datos que enlaza con el proyecto web que se está realizando. La implementación consistía simplemente en crear las tablas junto con sus atributos y una vez creadas todas comencé la relación entre tablas. Se tiene que tener en cuenta que las propiedades de cada atributo deben ser correctas, con esto me refiero a propiedades como:

- **NULL:** atributos que no pueden estar vacíos, como principalmente las claves primarias o información que siempre deba contener la tabla.
- **INT:** referente a atributos numéricos.
- **VARCHAR:** principalmente texto corto como títulos.
- **TEXT:** contenido de texto amplio como puede ser la descripción.
- **DATETIME:** para guardar información de fechas como la creación de un usuario o una sala.
- **AUTOINCREMENT:** es una propiedad que permite a un atributo autoincrementarse cada vez que un nuevo dato es introducido, muy útil para la secuencia de creaciones de id.

Una de las cosas a tener en cuenta de mayor importancia a la hora de la creación de una base de datos son las relaciones entre tablas, esto es muy importante debido a que muchas tablas tienen dependencias de otras para poder relacionar diferentes datos entre tablas, un ejemplo de porque debe crearse una buena relación entre tablas seria la eliminación de un dato y su consecuencia de eliminación en cascada, por ejemplo si yo elimino un usuario de la base de datos como consecuencia también debería eliminarse las salas que este usuario tiene creadas y todas sus amistades.

vista-controlador) el modelo es la parte que interactúa con la base de datos para obtener y tratar la información contenida en esta. La forma en que consigue estos datos es mediante sentencias sql.

7.1 Análisis de los modelos.

Los modelos van muy ligados a los controladores ya que estos son los que hacen uso de la información obtenida por el modelo para después pasársela a la vista, por eso ha sido necesario la comunicación entre los diferentes miembros del equipo para poder conocer que modelos desarrollar y que funciones de recuperación contendrá cada uno para poderlos separarlos y así garantizar que únicamente se puede obtener la información necesaria en cada sección de la aplicación sin tener que tener acceso a información irrelevante mejorando así la seguridad.

7.2 Conexión a la BBDD y implementación de modelos.

Lo primero que tenía que conseguir era realizar la conexión con la base de datos, realmente es una tarea sencilla ya que CODEIGNITER facilita mucho la tarea en lo referente a la conexión y manipulación de sentencia para la base de datos como veremos mas adelante.

- Configuración archivo "database"(Figura 3)

```
defined('BASEPATH') OR exit('No direct script access allowed');

$active_group = "default";
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'tandem';
$db['default']['dbdriver'] = 'mysqli';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8mb4_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;
```

Figura 3

En la parte que refiere a las sentencias sql(Figura 4) de los modelos gracias a CODEIGNITER la implementación ha sido mucho más sencilla y estructurada a los que estaba acostumbrado a hacer en PHP normal.

7 DESARROLLO MODELO

En el patrón de arquitectura de software MVC(modelo-

```

function getRoomInfo($session)
{
    $this->db->select('room.* , room_category.name_catRoom, room_avatar.name_img');
    $this->db->from('room');
    $this->db->join('rel_room_user', 'room.id_room=rel_room_user.id_room');
    $this->db->join('room_category', 'room.id_catRoom=room_category.id_catRoom');
    $this->db->join('room_avatar', 'room.id_avatar=room_avatar.id_avatar');
    $this->db->here('rel_room_user.id_user', $session);
    $query = $this->db->get();
    $query2 = $this->db->select('name_language');
    $this->db->from('languages');
    $this->db->here('languages.id_language', $query->row()->id_speak);
    $query2 = $this->db->get();
    $this->db->select('name_language');
    $this->db->from('languages');
    $this->db->here('languages.id_language', $query->row()->id_learn);
    $query3 = $this->db->get();

    $data = array(
        'title' => $query->row()->title,
        'name_catRoom' => $query->row()->name_catRoom,
        'description' => $query->row()->description,
        'capacity' => $query->row()->capacity,
        'speak' => $query2->row()->name_language,
        'learn' => $query3->row()->name_language,
        'avatar' => $query->row()->name_img,
        'connected' => $query->row()->connected
    );
    return $data;
}

```

Figura 4

Como se puede ver la estructura es muy sencilla, cada parte de la sentencia sql esta separa así facilitando la creación de las querys.

- **Select:** donde se seleccionaran los campos a obtener.
- **From:** las tablas a seleccionar
- **Join:** la unión de diferentes tablas
- **Where:** el condicional para filtrar.

8 DESARROLLO SECCIÓN ACADEMIAS.

Esta será la parte más importante en lo referente a las academias ya que aquí podrán modificar su información a mostrar mediante un template.

8.1 Análisis de información.

Ya recopilada toda la información de nuestros principales clientes, las academias, pude hacerme una idea lo que haría falta que tuviera la sección referente a las academias, para ello fue muy importante el haber tenido reuniones con las academias ya que por ejemplo pudimos conocer la importancia de que las academias tuvieran una landing page donde mostrar su información.

8.2 Estructuración de la sección.

Como ya he comentado anteriormente las academias tendrán acceso a la aplicación una vez nosotros le proporcionemos un usuario de entrada, para evitar tener academias a las cuales no estamos asociadas. Cuando ya tienen el usuario lo primero que deben hacer es proceder a llenar la información necesaria mediante un formulario:

- Información personal:
 - Nombre academia
 - Logotipo
 - Teléfono
 - Dirección

Una vez introducido la información procederán a modificar el template elegido, para ello se creara una estructura simple donde directamente podrán visualizar y editar la información directamente en el mismo template, de esta forma se consigue que la academia pueda ver como va quedando la información introducida directamente.

En la sección 8 de resultados podremos ver como ha quedado esta sección.

9 RESULTADOS

Primeramente decir que los resultados han sido favorables y bastante completos en respecto a lo que yo tenía que desarrollar, cierto es que falta definir alguna sección algo mejor y mejorar ideas en cara a líneas futuras (sección 10) y solucionar o mejorar secciones en las que hemos tenido problemas como explicaremos en el punto 11 "problemas y dificultades".

En esta sección iré mostrando parte por parte los resultados obtenidos del trabajo realizado para poder seguir un orden.

9.1 Resultados Base de datos

La base de datos es uno de los puntos mas importantes de todo el proyecto ya que al ser el núcleo todo debe estar perfecto para poder realizar un correcto desarrollo tanto en el fron-end como en el BackEnd/Backoffice y recuperar los datos necesarios para su modelación.

Finalmente la base de datos consistía de 14 tablas(figura 5) totalmente relacionadas y con las características de los atributos bien configuradas.

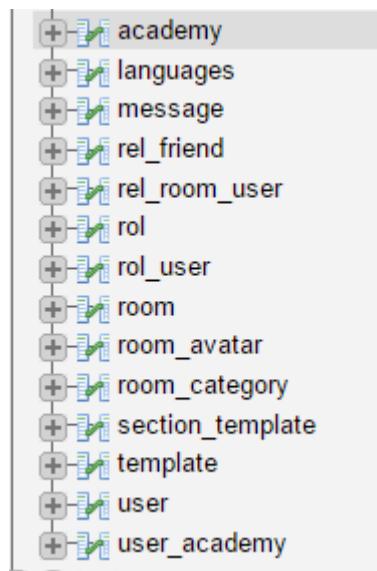


Figura 5

Al no poder mostrar todo el contenido os mostrare 2 ta-

blas bastante importantes de cada parte de la base de datos(user-room y academy)

- Tabla user (Figura 6)

| # | Nombre | Tipo | Cotejamiento |
|----|-----------------|--------------|--------------------|
| 1 | <u>id_user</u> | int(99) | |
| 2 | <u>username</u> | varchar(55) | utf8mb4_general_ci |
| 3 | <u>password</u> | varchar(100) | utf8mb4_general_ci |
| 4 | <u>mail</u> | varchar(55) | utf8mb4_general_ci |
| 5 | <u>created</u> | datetime | |
| 6 | <u>name</u> | varchar(55) | utf8mb4_general_ci |
| 7 | <u>surname</u> | varchar(55) | utf8mb4_general_ci |
| 8 | <u>gender</u> | int(1) | |
| 9 | <u>country</u> | varchar(55) | utf8mb4_general_ci |
| 10 | <u>avatar</u> | varchar(55) | utf8mb4_general_ci |

Figura 6

Una pequeña descripción de la tabla user sería que es utilizada para guardar todos los datos de los usuarios, pero realmente la parte más importante de esta tablas son sus relaciones ya que un usuario esta relacionado con las salas que se crearan y al mismo tiempo con idiomas, roles, amistades, ahora definiré un poco más las relaciones de esta tabla.

- **message:** para poder saber que mensaje pertenece a cada usuario.
- **Rel_friend:** relación para poder crear una amistad entre dos usuarios.
- **Rel_room_user:** tabla intermedia que relacionara las salas con el usuario al que le pertenece, esta tabla también esta relacionada con "room".
- **Rol_user:** tabla que relacionara los usuarios con los roles, es una tabla intermedia ya que un usuario en un futuro podrá tener mas de un rol.

La siguiente tabla a comentar es Academy(figura 8) parte importante en las academias, esta tabla tendrá toda la información personal de las academias, pero como en la anterior lo importante son las relaciones que ahora explicare.

| # | Nombre | Tipo | Cotejamiento |
|---|----------------------|-------------|--------------------|
| 1 | <u>id_academy</u> | int(255) | |
| 2 | <u>name</u> | varchar(55) | utf8mb4_general_ci |
| 3 | <u>phone</u> | int(12) | |
| 4 | <u>mail</u> | varchar(55) | utf8mb4_general_ci |
| 5 | <u>location</u> | varchar(55) | utf8mb4_general_ci |
| 6 | <u>cp</u> | int(10) | |
| 7 | <u>template</u> | int(1) | |
| 8 | <u>info_created</u> | int(1) | |
| 9 | <u>template_edit</u> | int(1) | |

Figura 7

- **Template:** se relacionara para conocer el template que tiene asignado la academia, esta tabla tambien se relacionara con section_template para poder configurar cada sección de la landing page de las academias.
- **User_academy:** se relacionara para conocer el usuario asignado a esa academia con el que deberán entrar para poder configurarla información.
- **Section_template:** igual que la relación de "template" se deberá relacionar con "academy" para conocer a quien pertenece la información a mostrar.

9.2 Resultados del modelo(MVC)

Mostrar resultados de esta sección es un poco complicado ya que se basa casi todo en código y funciones con sentencias sql, por ello explicare los modelos(figura 8) que tenemos y su función principal.

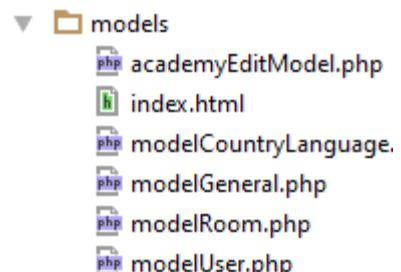


Figura 8

- **AcademyEditModel:** Este modelo contiene todos los update y inserts así como la recogida de información de la base de datos para que las academias puedan moldear su información
- **modelGeneral:** contiene sentencias que recuperan información general que no corresponde a una sección específica, como puede ser información a mostrar en una landing page.
- **modelCountryLanguage:** se utilizará para recuperar los idiomas y países de los usuarios, la idea principal es que en un futuro seamos capaces de reconocer la entrada de un usuario mediante geolocalización facilitando así la traducción directa de la aplicación.
- **modelRoom:** modelo bastante importante donde servirá para la recuperación y generación de las diferentes salas, tiene bastantes funciones que filtran como por ejemplo salas de amigos, salas filtradas por idiomas, etc.
- **modelUser:** modelo que se utilizará para recuperar toda la información de los usuarios y poder moldearla.

9.3 Resultados sección academias

Los clientes principales, las academias, esta sección es de vital importancia ya que en un futuro serán de ellos los que nos guiaran para poder crecer en la parte de negocio

y poder llegar a desarrollar una buena empresa, para ello es importante que tengan una sección propia donde puedan gestionar su información, cabe decir que a esta sección aún le faltan bastantes líneas futuras(sección 10) ya que aún tenemos que definir más todavía el modelo de negocio y reunirnos con más academias para acabar de definir el rumbo.

- Lo primero que deberán realizar las academias, una vez le facilitemos el usuario de entrada(Figura 8), es modificar sus datos a mostrar como academia, para ello he desarrollado un formulario(Figura 9) con diferentes pasos a seguir.

Figura 9

Una vez accedido si es la primera vez que entra saldrá el formulario para llenar la información.

Figura 10

- Una vez rellenada la información lo siguiente que deberá hacer la academia es modificar el template(Figura 11). Esta parte es muy intuitiva y tiene diferentes opciones que procedo a explicar por pasos.
 - Primer paso modificar el contenido, para ello todo el contenido modificable a sido introducido en cajas(Figura 12) de texto para la modificación en vivo.

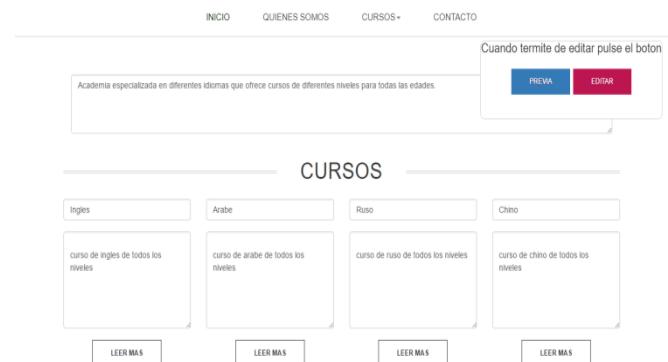


Figura 11

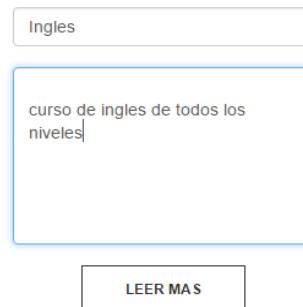


Figura 12

Cuando ya se ha acabado de modificar el contenido tendrá dos opciones(Figura 13) donde podrán o bien visualizar una vista previo o bien dar por finalizado.



Figura 13

Todo esto es para la primera vez que entra una academia a la aplicación, si en algún momento desean modificar sus datos se les habilitara el mismo proceso para poder modificarlos, lo único que ya le saldrá completada la información introducida.

10 PROBLEMAS.

Durante el desarrollo del proyecto han surgido diferentes problemas en el desarrollo ya sea por falta de conocimiento del Framework CODEIGNITER o diferentes problemas surgidos durante la implementación de código y funciones que nos ha costado solucionarlos. Aparte de estos problemas uno de los mas importantes ha sido la implementación de HANGOUTS, ha sido difícil poder

implementarlo al gusto y aún nos falta configurar opciones de capacidad y acceso, en un futuro tenemos que solventar estos problemas para la creación del negocio.

11 CONCLUSIONES.

Como conclusión podríamos decir que hemos conseguido un correcto MVP(MINIM VALUE PRODUCT) para enseñar a los clientes y poder comenzar a captar ideas para el desarrollo de la empresa y si es posible ya clientes que puedan estar asociados a nosotros en el momento que lancemos la aplicación web.

Por lo que ha mi parte respecta creo que he conseguido finalizar la mayoría de los puntos necesarios, pudiendo completar correctamente los puntos importantes y con un buen nivel de detalle y estoy contento con el trabajo realizado y sobre todo con el trabajo en equipo

Una de las cosas más importantes que he aprendido ha sido al usar el Framework CODEIGNITER, nunca había utilizado un Framework para la programación y la verdad es que es de mucha utilidad ya que a la hora de estructurar el MVC a sido clave en algunos aspectos para seguir esta arquitectura al pie de la letra, otra cosa que me a sorprendido de este Framework es la facilidad que ofrece a la hora de utilización de sentencia PHP y SQL facilitando el tratamiento de las funciones que normalmente me tomarían varias líneas completar a unas pocas.

Por otra parte me ha parecido interesante y cercano a la vida real el trabajar en equipo, realmente es mas difícil de lo que parece coordinarse, se tiene que tener todo muy coordinado y poder realmente mantener la comunicación con los demás miembros del equipo para que el proyecto pueda salir adelante sin dificultades, al principio nos costó adaptarnos un poco a las metodologías utilizadas como scrum, pero realmente son muy útiles para este tipo de proyecto el uso de sprints.

aspectos que pueden influir en el contenido de la base de datos.

Tenemos que tener en cuenta que para la futura implementación del negocio puede variar el modelo de negocio gracias a la participación de las academias, por ello puedes suceder bastantes cambios sobre todo en lo referente a la sección y funcionalidades de las academias, son cosas a tener en cuenta para la estructuración futura.

Otro punto a tener en cuenta en el futuro es la interacción con redes sociales ya que son una herramienta de la que podemos lucrar fuertemente para dar a conocer nuestra aplicación.

Un punto que he comentado anteriormente sería la geolocalización para poder conocer de donde provienen los usuarios que se conectan para poder hacer la traducción automáticamente.

AGRADECIMIENTOS

Primero de todo quiero agradecer a nuestro tutor, Juanjo Villanueva, por la oportunidad de dejarnos realizar este proyecto y por todo el asesoramiento dado sobre todo en la parte de negocio, también quiero agradecer a mi familia por el soporte y compañeros de la universidad, y por ultimo agradecer a los miembros del equipo que formamos el proyecto por la dedicación.

BIBLIOGRAFÍA

- [1] Bootstrap [Online]. Available: <http://getbootstrap.com/>
- [2] CODEIGNITER manual [Online]. Available: http://www.codeigniter.com/user_guide/
- [3] PHPMYADMIN docs [Online]. Available: <http://docs.phpmyadmin.net/en/latest/>
- [4] SQL [Online] Available: <http://www.w3schools.com/sql>
- [5] HANGOUTS [Online] Available: <https://developers.google.com/+hangouts/>
- [6] Canvas [Online] Available: <http://innokabi.com/canvas-de-modo-de-negocio>
- [7] Start up [Online] Available : http://es.wikipedia.org/wiki/Compa%C3%B1a_startup

12 LÍNEAS FUTURAS.

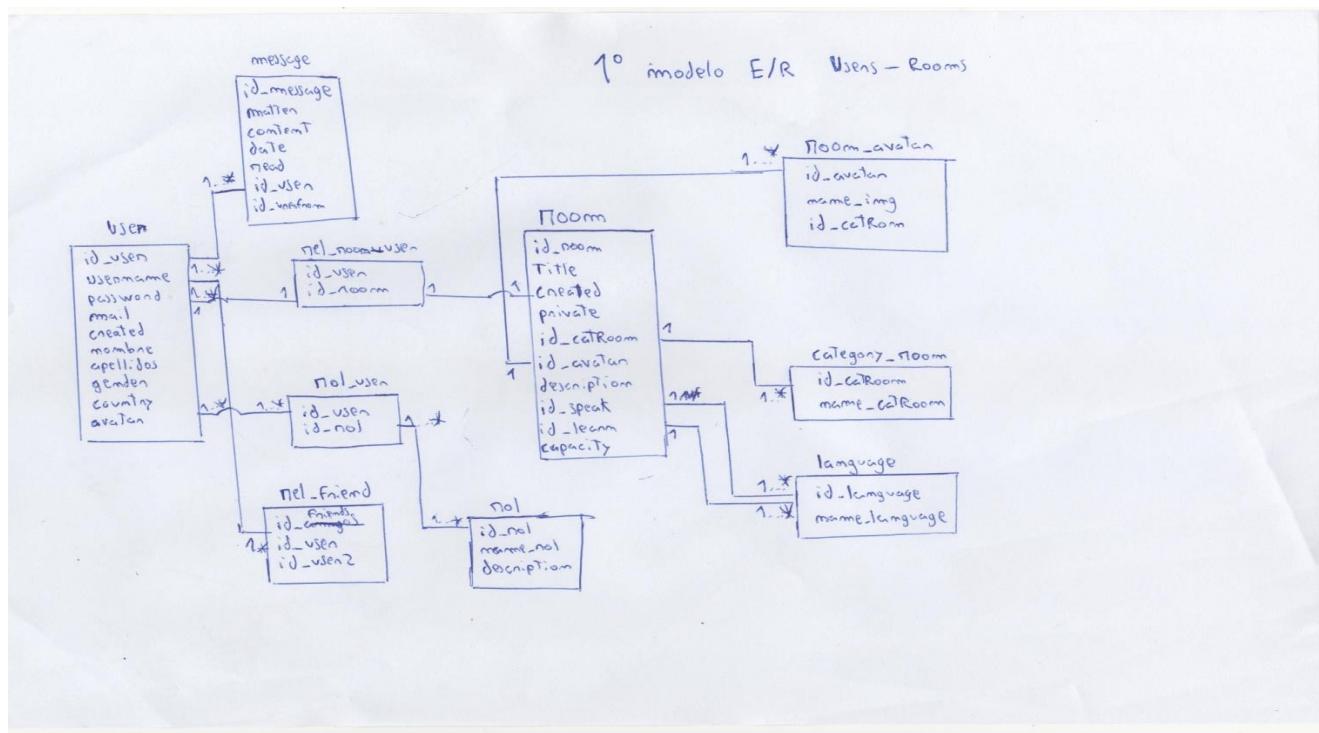
En un futuro se irán añadiendo más funcionalidades, como usuarios vip, diferentes roles de usuario, diferentes

APENDICE

A1. CANVAS

| Key Partners | Key Activities | Value Propositions | Customer 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. | Academias. Personas interesadas en aprender idiomas. |
| | Key Resources Servicio videoconferencia Foro. Blog. | Relaciones sociales, temas de interés. Cartera usuarios VIP Cartera Profesores | Channels Publicidad en academias Portal Web. | Profesores. Escuelas |
| | | | | Redes Sociales |
| Cost structure | Servidor web(Amazon/Azure) Dominio Personal | Revenue Streams Academias ofreciendoles una cartera de profesores + buscador Google AdWords Programa de afiliados. Sistema de cobro por clase impartidas mediante la aplicación | | |

A2. MODELO E/R USER-ROOM



A3. Modelo e/r Academias

