

***Títol:*** Didici - Language Learning Platform

***Volum:*** 1/1

***Alumne:*** Rubén Arroyo Gil

***Director:*** Leandro Navarro Moldes

***Departament:*** Arquitectura de Computadors

***Data:*** Juny 2012







---

## DADES DEL PROJECTE

*Títol del projecte:* Didici - Language Learning Platform

*Nom de l'estudiant:* Rubén Arroyo Gil

*Titulació:* Enginyeria Superior en Informàtica

*Crèdits:* 37,5

*Director:* Leandro Navarro Moldes

*Departament:* Arquitectura de Computadors

---

## MEMBRES DEL TRIBUNAL *(nom i signatura)*

*President:* Felix Freitag

*Vocal:* Maria Teresa Abad Soriano

*Secretari:* Leandro Navarro Moldes

---

## QUALIFICACIÓ

*Qualificació numèrica:*

*Qualificació descriptiva:*

*Data:*

---









# Didici - Language Learning Platform

Rubén Arroyo Gil

Juny 2012



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Situación actual . . . . .	1
1.2. Motivación . . . . .	1
1.3. Objetivos . . . . .	2
1.4. Procesos necesarios . . . . .	2
1.5. Planificación inicial . . . . .	2
<b>2. Análisis de requisitos</b>	<b>5</b>
2.1. Ingeniería de requisitos . . . . .	5
2.2. Método Volere . . . . .	5
2.2.1. Convenciones de nombres . . . . .	6
2.2.2. Propósito del proyecto . . . . .	6
2.2.3. Usuarios de la aplicación . . . . .	6
2.2.4. Restricciones obligatorias . . . . .	6
2.2.5. Ámbito del producto . . . . .	6
2.2.6. Requisitos funcionales . . . . .	7
2.2.7. Requisitos no funcionales . . . . .	7
2.2.8. Riesgos . . . . .	8
2.2.9. Documentación de usuario . . . . .	8
<b>3. Especificación</b>	<b>9</b>
3.1. Modelo de Casos de Uso . . . . .	9
3.1.1. Especificación de los Casos de Uso . . . . .	10
3.2. Modelo conceptual . . . . .	23
3.2.1. Diagrama de clases . . . . .	24
3.2.2. Claves externas y restricciones textuales . . . . .	25
<b>4. Diseño</b>	<b>27</b>
4.1. Decisiones de diseño . . . . .	27
4.1.1. Plataforma física . . . . .	27
4.1.2. Modelo de diseño de la aplicación . . . . .	27
4.1.3. Capa de presentación . . . . .	28
4.1.4. Capa de dominio . . . . .	29
4.1.5. Capa de acceso a datos . . . . .	29
4.2. Diseño de la arquitectura del sistema . . . . .	29
4.2.1. Diseño de la capa de presentación . . . . .	29
4.2.2. Diseño de la capa de dominio . . . . .	34
4.2.3. Diseño de la capa de acceso a datos . . . . .	35

<b>5. Implementación</b>	<b>39</b>
5.1. Estrategia de desarrollo	39
5.2. Tecnologías utilizadas (LAMP)	39
5.3. Entorno de desarrollo	40
5.4. Lenguajes de programación utilizados	40
5.4.1. HTML	40
5.4.2. CSS	41
5.4.3. PHP	41
5.4.4. JavaScript	42
5.4.5. Java	42
5.5. Integración de plugins y otras herramientas	43
5.5.1. TFPDF	43
5.5.2. Gallerific	43
5.5.3. Flowplayer	43
5.5.4. upload.class.php	44
5.5.5. datatables	44
<b>6. Pruebas</b>	<b>45</b>
6.1. Prueba modular	45
6.2. Prueba en desarrollo	45
6.3. Prueba global de la aplicación	46
<b>7. Conclusión</b>	<b>47</b>
7.1. Estudio de la planificación	47
7.2. Estudio económico	49
7.2.1. Coste de desarrollo	49
7.2.2. Coste de implantación	49
7.2.3. Recursos humanos	49
7.3. Objetivos conseguidos	50
7.4. Líneas de futuro	51
7.5. Conocimientos obtenidos	52
7.6. Valoración personal	52
<b>A. Análisis de las iteraciones</b>	<b>53</b>
<b>B. Vista final de la aplicación</b>	<b>55</b>
<b>Bibliografía</b>	<b>66</b>
<b>Glosario</b>	<b>68</b>

# Índice de figuras

3.1. Actores que intervienen en el sistema y casos de uso más significativos. . . . .	10
3.2. Diagrama de los casos de uso Login y Logout. . . . .	10
3.3. Diagrama de los casos de uso Crear, Actualizar y Eliminar Usuario. . . . .	12
3.4. Diagrama de los casos de uso Crear, Actualizar y Eliminar Categoría. . . . .	13
3.5. Diagrama de los casos de uso Añadir y Eliminar Imagen. . . . .	15
3.6. Diagrama de los casos de uso Crear, Actualizar y Eliminar Item Traducible. . . .	20
3.7. Diagrama de los casos de uso Crear, Actualizar y Eliminar Traducción de Item Traducible. . . . .	20
3.8. Diagrama de los casos de uso Generar presentación web. . . . .	21
3.9. Diagrama de los casos de uso Generar ejercicios interactivos. . . . .	22
3.10. Diagrama de los casos de uso Generar PDF. . . . .	23
3.11. Diagrama de classes. . . . .	24
4.1. Arquitectura de la aplicación. . . . .	28
4.2. Diseño de la pantalla que muestra presentaciones. . . . .	30
4.3. Diseño de la pantalla que muestra los ejercicios. Dos tipos de ejercicio. . . . .	30
4.4. Diseño de la pantalla que muestra los ejercicios. Otros dos tipos de ejercicio. . . .	31
4.5. Diseño de la pantalla para registrar un nuevo ítem. . . . .	32
4.6. Diseño de la pantalla que muestra la zona de traducción. . . . .	34
4.7. Diseño de los controladores del sistema. . . . .	35
4.8. Objeto DAO. . . . .	36
5.1. Ciclo de vida del desarrollo iterativo e incremental . . . . .	39
A.1. Diagrama de clases de la primera y segunda iteración . . . . .	54
B.1. Vista de la <i>zona de imágenes</i> . . . . .	55
B.2. Vista del formulario <i>nuevo ítem</i> . . . . .	56
B.3. Vista de la <i>zona de traducción</i> . . . . .	57
B.4. Vista principal de la aplicación. . . . .	58
B.5. Vista de los ejercicios interactivos. Ejercicio tipo <i>rellenar huecos (Filling the gaps)</i> . . . . .	59
B.6. Vista de los ejercicios interactivos. Ejercicio tipo <i>Ordenar letras (Sort letters)</i> . . .	60
B.7. Vista de los ejercicios interactivos. Ejercicio tipo <i>Ordenar frase (Sort sentence)</i> . .	61
B.8. Vista de los ejercicios interactivos. Ejercicio tipo <i>Ordenar frases (Sort sentences)</i> . .	62
B.9. Vista de los <i>ejercicios en PDF</i> . . . . .	63



# Capítulo 1

## Introducción

Si decidimos aprender idiomas a través de Internet nos toparemos con infinidad de páginas web que muestran el contenido estructurado en lecciones, cada una de ellas con una serie de expresiones en un idioma. Algunos portales muestran estas expresiones traducidas en otro idioma, otros disponen de ejercicios relativamente estáticos, en los que se interactúa con las expresiones en el idioma que aprendemos.

Conseguir guardar de forma eficiente la información para luego recuperarla de forma más efectiva es lo que se puede esperar de cualquier software. En el caso de las plataformas para aprender idiomas, si almacenamos una serie de expresiones traducidas en diferentes idiomas, disponemos de un gran potencial para realizar diferentes ejercicios en todos estos idiomas. Además, si contamos con una representación gráfica y un registro de voz para cada elemento guardado, se incrementan las posibilidades de crear situaciones de aprendizaje de forma automatizada.

### 1.1. Situación actual

Algunos de los gestores de contenidos de ámbito general y distribución libre pueden ser Joomla o Drupal, mientras que Moodle se presenta como un gestor de cursos para ayudar a educadores a crear comunidades de aprendizaje.

Existen infinidad de cursos de idiomas en Internet y también existen plataformas dedicadas específicamente a la enseñanza de idiomas como pueden ser Babbel, Rosseta Stone o Livemocha.

En el glosario de este documento se puede encontrar una pequeña descripción de cada una de las plataformas mencionadas con anterioridad.

### 1.2. Motivación

Las plataformas actuales no permiten el acceso rápido a la información y esta está muchas veces organizada de una forma lineal, existiendo un único flujo de acceso a una misma información. Esto normalmente no permite al usuario cambiar de forma ágil el contenido que quiere mostrar.

Almacenando y presentando de forma eficiente la información se puede obtener un mayor rendimiento en cuestiones de usabilidad de la aplicación, obteniendo diferentes escenas de aprendizaje a partir de un mismo concepto.

### 1.3. Objetivos

El objetivo del proyecto es construir una herramienta para la gestión de contenidos didácticos, especialmente enfocada a la enseñanza de idiomas.

Almacenando frase, imagen y audio, para un mismo concepto en diferentes idiomas, los usuarios podrán generar situaciones de aprendizaje de forma automática o semi-automática.

El objetivo principal, la generación automática de ejercicios, se logrará mediante el almacenamiento y la obtención eficiente de la información que previamente categorizada podrá ser accedida de forma rápida e intuitiva por el usuario.

### 1.4. Procesos necesarios

- Recopilación de información sobre los gestores de contenido actuales.
- Análisis de los requisitos de la plataforma.
- Especificación del modelo de la plataforma.
- Especificación de la interfaz.
- Especificación de los casos de uso del sistema.
- Selección de la tecnologías utilizadas para la implementación.
- Familiarización con los lenguajes de programación necesarios para el desarrollo de la plataforma.
- Diseño e implementación de la plataforma.
- Diseño e implementación de la interfaz.
- Realización de pruebas de consistencia de la aplicación.

### 1.5. Planificación inicial

Tarea	Horas	Días
Descripción del proyecto y objetivos	40	10
Análisis de requisitos	80	20
Especificación	160	40
Diseño	200	50
Aprendizaje de las tecnologías	100	25
Implementación	240	60
Pruebas	20	5
Documentación	60	15
<b>Total</b>	<b>900</b>	<b>225</b>



**Descripción del proyecto y objetivos** Se exponen las ideas generales del proyecto que pasarán a definir los objetivos del mismo.

**Análisis de requisitos** Una vez definidos los objetivos se definirán las necesidades del usuario final que pasarán a ser los requisitos del sistema.

**Especificación** La especificación es la base donde se sustentará el sistema, en este caso un sistema software. Esta fase tendrá una duración de 40 días.

**Diseño** La fase de diseño puede realizarse sin haber acabado la fase de especificación, pues existirán partes diferenciadas en el proyecto que permitirán ese paralelismo.

**Aprendizaje de las tecnologías** Antes de pasar a la siguiente etapa y una vez elegidas las tecnologías con las que se desarrollarán cada una de las partes del proyecto resultantes del diseño, es necesario realizar un estudio previo para valorar las tecnologías y realizar pequeños prototipos. En esta fase se puede llegar a elegir una nueva tecnología en detrimento de otra de la que no se obtenga el resultado esperado.

**Implementación** El tiempo estimado de duración de la implementación del proyecto es de 60 días, pudiendo variar en función de las dificultades que se presenten en la integración de tecnologías.

**Pruebas** Una vez finalizada la implementación es indispensable una fase de pruebas en la que se analiza de forma conjunta el funcionamiento integral de la aplicación, normalmente se identifican y resuelven los bugs de la aplicación.

**Documentación** Como último proceso se realiza la documentación del proyecto, explicando como se ha conseguido el desarrollo del software y haciendo un análisis técnico de cada una de las etapas necesarias para la realización de la aplicación.



## Capítulo 2

# Análisis de requisitos

En este capítulo se describen los requisitos necesarios para el sistema. Se entiende por requisito la capacidad o condición necesaria por parte del usuario para solucionar un problema o alcanzar un objetivo. El objetivo de esta fase es, por tanto, conocer las funcionalidades y cualidades que debe tener el sistema.

### 2.1. Ingeniería de requisitos

El software tiene que satisfacer varios requisitos generales. En primer lugar, tiene que seguir una estructura modular, dividiendo lógicamente el sistema, facilitando su comprensión, el mantenimiento y la ampliabilidad.

La introducción de comentarios en el código ayuda a una mejor comprensión y mayor velocidad de mantenimiento y ampliabilidad. Finalmente debe incluir una interfaz clara, sencilla y con mensajes de error comprensibles.

A parte de los generales, los requisitos del software pueden ser también funcionales y no funcionales.

La ingeniería de requisitos es una parte muy importante en el proceso de ingeniería del software, donde analistas y desarrolladores identifican las necesidades y requisitos del software que pretenden desarrollar. Conociendo las necesidades se puede modelar y diseñar una solución que cumpla todas ellas.

### 2.2. Método Volere

El *método Volere* es el resultado de diversos años de práctica, consultoría e investigación en la ingeniería de requisitos. Suzanne Robertson y James Robertson han documentado la experiencia obtenida durante años especificando una plantilla que proporciona ayuda para el inicio de nuevos proyectos de software. [2]

Adaptando y aplicando esta plantilla a las necesidades del proyecto se obtiene la información necesaria para ser el punto de partida del mismo. A continuación se muestra el resultado del pre-análisis aplicando el *método Volere*.

### 2.2.1. Convenciones de nombres

A continuación se muestra un glosario con algunos de los nombres utilizados en este documento.

**Superusuario** Persona que tiene acceso a la gestión integral de la aplicación.

**Editor** Persona o proceso que tiene acceso a la parte administrativa de la aplicación.

**Invitado** Persona o proceso que tiene acceso a la información generada por la aplicación.

**Ítem Traducible, Expresión o Frase** Texto registrado en el sistema que puede tener traducciones a diferentes idiomas.

**Ejercicio** Conjunto de ítems traducibles dispuestos de una forma específica.

### 2.2.2. Propósito del proyecto

El propósito de este proyecto es dar soporte al aprendizaje de lenguas extranjeras generando situaciones de aprendizaje a partir de información previamente introducida. La funcionalidad principal deseada es conseguir la automatización de ejercicios a partir de expresiones introducidas en el sistema.

### 2.2.3. Usuarios de la aplicación

Existen tres tipos de usuarios

- Superusuario.
- Editor.
- Invitado.

El *Superusuario* tiene acceso a todas las funcionalidades del sistema, mientras que el *Editor* puede realizar las tareas de edición en una serie de idiomas especificados por el *Superusuario*. Por otro lado, el usuario *Invitado* tiene acceso a la interacción con los contenidos, pudiendo consultar la información introducida y realizar ejercicios.

### 2.2.4. Restricciones obligatorias

- Todas las herramientas utilizadas para la creación del sistema deben ser Open Source. [3]
- La gestión de la herramienta se realiza mediante Internet. Cualquier usuario con acceso a Internet y un navegador web podrá utilizar la herramienta.

### Entorno de implementación

Para asegurar una mayor disponibilidad de la herramienta es recomendable instalar el sistema en proveedores especializados. Esto no impide que el sistema se pueda instalar en un ordenador común, siempre disponiendo de un servidor de aplicaciones.

### 2.2.5. Ámbito del producto

Este punto define los casos de uso del sistema, estos se especificarán en el siguiente capítulo.

### 2.2.6. Requisitos funcionales

Los requisitos funcionales describen las funcionalidades que debe tener nuestro sistema. Estos son los que se consideran para la realización del sistema.

- La gestión de contenidos debe poder realizarse desde cualquier parte del mundo mediante una conexión a Internet y un navegador web.
- Los contenidos deben estar organizados de una forma intuitiva y deben motivar al usuario a utilizar la aplicación.
- Los contenidos publicados deben acabar siendo páginas web accesibles por los usuarios que visiten el portal.
- El sistema debe diferenciar claramente la gestión por parte del administrador y la interacción por parte del usuario.
- El sistema dispone de un conjunto de idiomas predeterminado, correspondientes a los incluidos en la norma ISO639-2 [4]
- La herramienta debe permitir crear expresiones (ítems traducibles) en cualquiera de los idiomas disponibles. Estos ítems podrán ser posteriormente actualizados y/o eliminados.
- Los ítems traducibles podrán ser categorizados según sean conceptos de vocabulario o frases.
- Un ítem traducible puede ser traducido a todos los idiomas presentes en el sistema.
- Las expresiones pueden ser categorizadas. Las categorías creadas en el sistema pueden a su vez ser traducidas a cualquier otro idioma presente en el sistema. Una vez creadas, tanto expresiones como categorías podrán ser actualizadas o eliminadas.
- El sistema debe permitir la visualización de expresiones de manera interactiva.
- El sistema debe ser capaz de crear diferentes tipos de ejercicios a partir de los contenidos introducidos.
- Los ejercicios creados por el sistema deben ser dinámicos e interactivos.
- El usuario tendrá también la opción de generar ejercicios para imprimir, los cuales estarán adaptados para poder ser realizados sobre papel y se generarán en formato PDF.

### 2.2.7. Requisitos no funcionales

#### Requisitos de apariencia

La interfaz debe ser atractiva para el usuario, organizando los contenidos de manera intuitiva. El usuario debe sentirse cómodo utilizando el software.

#### Requisitos de usabilidad

La parte de administración debe ser clara e intuitiva. Se usarán iconos y descripciones para hacer que la interfaz sea más amigable.

#### Requisitos operacionales

La herramienta podrá ser instalada en un servidor de páginas web.

### Requisitos de seguridad

La parte administrativa debe ser accesible por un conjunto limitado de usuarios. El *Super-usuario* es el encargado de asignar nuevos editores.

Los *editores* tendrán acceso restringido a ciertas funcionalidades administrativas.

Dependiendo de los datos confidenciales que se registren se deberá utilizar un fichero de protección de datos.

### Requisitos políticos y culturales

Al ser un sistema multilingüe, se requiere que la interfaz pueda ser traducida a cualquier idioma presente en el sistema. Del mismo modo un usuario podrá elegir el idioma en el que se le presenta la interfaz.

### Requisitos legales

Dependiendo de los datos de usuario que registremos necesitaremos un fichero automatizado que cumpla la Ley Orgánica de Protección de Datos (LOPD)[5]

#### 2.2.8. Riesgos

La flexibilidad a la hora de crear expresiones y sus traducciones a un gran número de idiomas puede dar lugar a que existan traducciones que si bien en un idioma son correctas, en otro puedan ser mejorables.

#### 2.2.9. Documentación de usuario

El sistema debe ser suficientemente sencillo como para no requerir de un manual de usuario, pero se mostrarán mensajes informativos en momentos específicos que ayudarán al usuario a utilizar la aplicación.

## Capítulo 3

# Especificación

Gracias a la especificación de software podemos definir que queremos que haga nuestro sistema con independencia de como lo haga. Para la especificación del sistema necesitamos un lenguaje de modelado, en nuestro caso utilizaremos Unified Modeling Language (UML). Se utilizará UML en la especificación por ser un estándar en la construcción de software.

Para la realización de la especificación se ha utilizado Dia Diagram Editor.[6]

El lenguaje UML permite crear modelos previos a la construcción de un sistema software asegurando la solidez de su arquitectura. Con proyectos de gran complejidad es difícil conocer la totalidad del sistema sin disponer de modelos que faciliten su comprensión.

El resultado de la especificación en UML:

- Modelo de Casos de Uso: define la interacción entre los actores y el sistema software.
- Modelo Conceptual: define los conceptos relevantes del mundo real de referencia.
- Diagramas de secuencia: describe la respuesta del sistema a eventos externos.
- Contratos de las operaciones: describe que hacen las operaciones del sistema.
- Diagrama de estados: describe que estados recorren los elementos del sistema.

### 3.1. Modelo de Casos de Uso

El Modelo de Casos de Uso se define a partir de los siguientes elementos fundamentales:

**Caso de uso:** describe la secuencia de eventos que un agente externo (actor) realiza en el sistema con el fin de alcanzar un objetivo.

**Actor:** entidad externa que ejecuta un caso de uso.

La Figura 3.1 muestra los actores que intervienen en el sistema y los casos de uso que se generan a partir de los requisitos expuestos anteriormente.

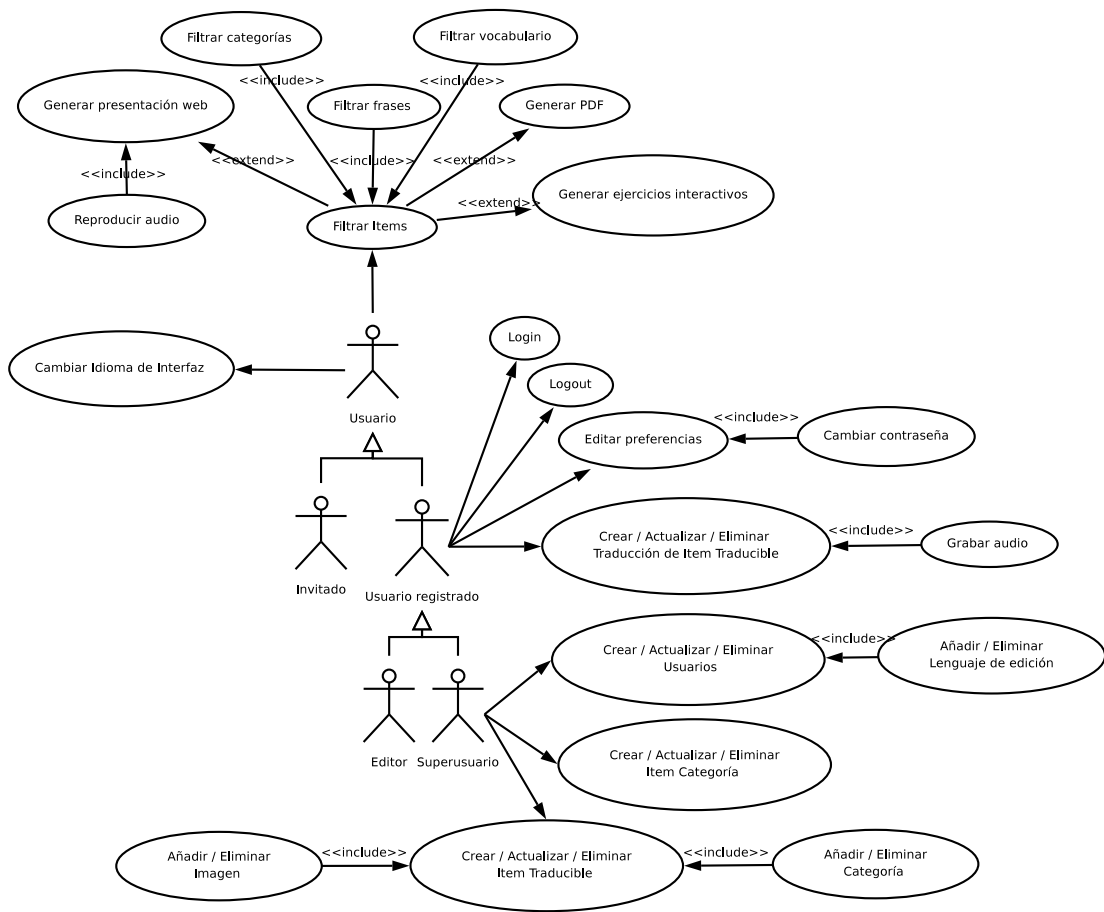


Figura 3.1: Actores que intervienen en el sistema y casos de uso más significativos.

### 3.1.1. Especificación de los Casos de Uso

A continuación se especifican algunos de los casos de uso más significativos de la plataforma.

#### Login y Logout

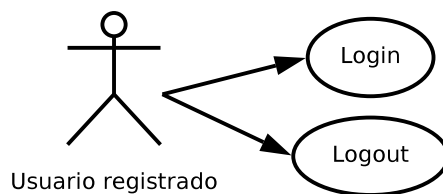


Figura 3.2: Diagrama de los casos de uso Login y Logout.



<b>1. Login</b>	
<b>Descripción</b>	
El Sistema debe permitir la gestión de los contenidos a usuarios registrados.	
<b>Actor principal</b>	Usuario registrado
<b>Precondición</b>	
-	
<b>Criterio de validación</b>	
El usuario accede a la zona de administración de la aplicación.	
<b>Curso típico de eventos</b>	
Usuario	Sistema
1. El usuario registrado decide acceder a la aplicación	
	2. El Sistema muestra los campos necesarios para validar el acceso a la plataforma. (nombre y contraseña)
3. El usuario registrado introduce su nombre y su contraseña.	
	4. El Sistema valida la información introducida por el usuario.
5. El usuario accede a la zona de administración de la plataforma.	
<b>Cursos alternativos</b>	
4a. No existe usuario o contraseña incorrecta.	
4a1. El Sistema avisa al usuario con un mensaje de error.	

<b>2. Logout</b>	
<b>Descripción</b>	
El sistema debe permitir a los usuarios logueados en la plataforma la posibilidad de desconectarse.	
<b>Actor principal</b>	Usuario Registrado
<b>Precondición</b>	
El Usuario está logueado en la plataforma	
<b>Criterio de validación</b>	
El Usuario Registrado abandona la zona de administración de la aplicación.	
<b>Curso típico de eventos</b>	
Usuario	Sistema
1. El Usuario Registrado decide abandonar la zona de administración de la aplicación	
	2. El Sistema desconecta al usuario de la aplicación y lo redirige hacia la zona pública.
<b>Cursos alternativos</b>	

Crear, Actualizar y Eliminar Usuario

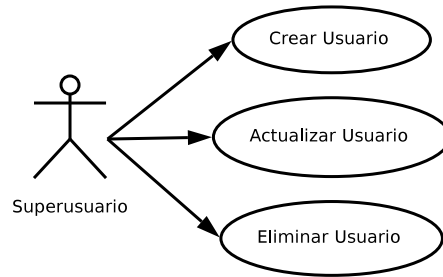


Figura 3.3: Diagrama de los casos de uso Crear, Actualizar y Eliminar Usuario.

### 3. Crear Usuario

#### Descripción

La herramienta debe permitir a Superusuarios el registro de Usuarios.

#### Actor principal

Superusuario

#### Precondición

-

#### Criterio de validación

El Superusuario crea un nuevo Usuario Registrado en el Sistema.

#### Curso típico de eventos

Usuario	Sistema
1. El Superusuario decide crear un nuevo Usuario	
	2. El Sistema muestra los campos necesarios para el nuevo usuario.
3. El Superusuario selecciona las opciones convenientes.	
	4. El Sistema crea el nuevo Usuario Registrado.

#### Cursos alternativos

3a. El Superusuario cancela la operación.

### 4. Actualizar Usuario

#### Descripción

Los Superusuarios pueden actualizar las opciones de edición de Usuarios Editores.

#### Actor principal

Superusuario

#### Precondición

-

#### Criterio de validación

Las opciones de edición del Usuario Editor seleccionado por el Superusuario se han actualizado correctamente.

#### Curso típico de eventos

Usuario	Sistema
1. El Superusuario decide actualizar las opciones de edición de un Usuario Editor.	
	2. El Sistema muestra al Superusuario todos los Usuarios Registrados en el Sistema.
3. El Superusuario selecciona el Usuario que desea actualizar.	
	4. El Sistema muestra los idiomas en los que el Usuario puede editar y permite su modificación.
5. El Superusuario realiza las modificaciones convenientes.	
	6. El Sistema actualiza el Usuario.

#### Cursos alternativos

5a. El Superusuario cancela la operación.

<b>5. Eliminar Usuario</b>	
<b>Descripción</b>	
Los Superusuarios pueden eliminar a Usuarios Registrados.	
<b>Actor principal</b>	Superusuario
<b>Precondición</b>	
-	
<b>Criterio de validación</b>	
El Usuario seleccionado por el Superusuario se ha eliminado del Sistema.	
<b>Curso típico de eventos</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El Superusuario decide eliminar un Usuario.	2. El Sistema muestra los Usuarios registrados hasta el momento.
3. El Superusuario selecciona el Usuario que desea eliminar.	4. El Sistema solicita la confirmación de la acción.
5. El Superusuario confirma la acción.	6. El Sistema elimina la referencia al Usuario.
<b>Cursos alternativos</b>	
5a. El Superusuario cancela la operación.	

**Crear, Actualizar y Eliminar Categoría**

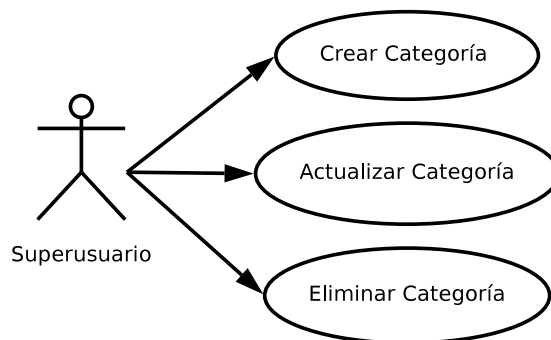


Figura 3.4: Diagrama de los casos de uso Crear, Actualizar y Eliminar Categoría.

<b>6. Crear Categoría</b>	
<b>Descripción</b>	
Las expresiones pueden ser categorizadas.	
<b>Actor principal</b>	Superusuario

<b>Precondición</b>	
-	
<b>Criterio de validación</b>	
La nueva categoría se ha registrado en el Sistema.	
<b>Curso típico de eventos</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El Superusuario decide crear una nueva categoría.	2. El Sistema muestra el formulario de creación de categorías en el idioma predeterminado.
3. El Superusuario completa la información referente a la nueva categoría (nombre de la categoría).	4. El Sistema guarda una referencia a la categoría creada por el Superusuario.
<b>Cursos alternativos</b>	
3a. El Superusuario cancela la operación.	

---

### 7. *Actualizar Categoría*

#### **Descripción**

Las categorías registradas en el Sistema pueden ser posteriormente actualizadas.

#### **Actor principal**

Superusuario

#### **Precondición**

-

#### **Criterio de validación**

Se actualiza el nombre de la categoría en el idioma predeterminado.

#### **Curso típico de eventos**

**Usuario**

**Sistema**

1. El Superusuario decide actualizar una Categoría.

2. El Sistema muestra las Categorías registradas en el Sistema en el idioma predeterminado.

3. El Superusuario selecciona la Categoría que desea actualizar y la modifica.

4. El Sistema guarda las modificaciones realizadas por el Superusuario.

#### **Cursos alternativos**

3a. El Superusuario cancela la operación.

### 8. *Eliminar Categoría*

#### **Descripción**

El Sistema debe permitir eliminar las categorías registradas.

#### **Actor principal**

Superusuario

<b>Precondición</b>	
-	
<b>Criterio de validación</b>	
La Categoría seleccionada se ha eliminado del Sistema.	
<b>Curso típico de eventos</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El Superusuario decide eliminar una Categoría.	2. El Sistema muestra las categorías registradas hasta el momento en el idioma predeterminado.
3. El Superusuario selecciona la Categoría que desea eliminar.	4. El Sistema solicita la confirmación de la acción.
5. El Superusuario confirma la acción.	6. El Sistema elimina la referencia a la Categoría seleccionada.
<b>Cursos alternativos</b>	
4a. El Superusuario cancela la operación.	

### Añadir y Eliminar Imagen

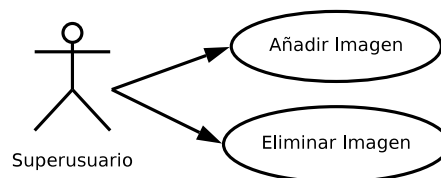


Figura 3.5: Diagrama de los casos de uso Añadir y Eliminar Imagen.

#### 9. Añadir Imagen

##### Descripción

La herramienta debe permitir introducir imágenes para ser posteriormente asociadas a las expresiones.

##### Actor principal

Superusuario

##### Precondición

-

##### Criterio de validación

La nueva imagen se ha registrado en el Sistema.

##### Curso típico de eventos

**Usuario**

**Sistema**

1. El Superusuario decide añadir una nueva imagen al Sistema.

2. El Sistema muestra el selector de archivos predeterminado del Sistema Operativo.

3. El Superusuario selecciona un archivo.
4. El Sistema comprueba que el archivo sea un archivo de Imagen y lo registra en el Sistema.

**Cursos alternativos**

- 4a. El archivo seleccionado no es un archivo de Imagen.
    - 4a1. El Sistema avisa al usuario con un mensaje de error.
- 

**10. Eliminar Imagen****Descripción**

Los Superusuarios pueden eliminar las imágenes registradas.

**Actor principal**

Superusuario

**Precondición**

-

**Criterio de validación**

La Imagen seleccionada por el Superusuario se ha eliminado del Sistema.

**Curso típico de eventos**

Usuario	Sistema
1. El Superusuario decide eliminar una Imagen.	
	2. El Sistema muestra los Imágenes registradas hasta el momento.
3. El Superusuario selecciona la Imagen que desea eliminar.	
	4. El Sistema solicita la confirmación de la acción.
5. El Superusuario confirma la acción.	
	6. El Sistema elimina la referencia a la Imagen.

**Cursos alternativos**

- 5a. El Superusuario cancela la operación.
- 

**Crear, Actualizar y Eliminar Item Traducible****11. Crear Ítem Traducible****Descripción**

La herramienta debe permitir crear expresiones (ítems traducibles) en cualquiera de los idiomas disponibles.

**Actor principal**

Superusuario

**Precondición**

-

**Criterio de validación**

El Superusuario crea un Ítem Traducible con la información introducida.

**Curso típico de eventos**

Usuario	Sistema
1. El Superusuario decide crear un Ítem Traducible	
	2. El Sistema muestra las categorías, las imágenes y los posibles tipos de ítem (frase o vocabulario) para ser relacionadas con el nuevo Ítem Traducible.
3. El Superusuario selecciona las opciones convenientes.	
	4. El Sistema crea el nuevo Ítem Traducible con las características definidas por el Superusuario.
<b>Cursos alternativos</b>	
3a. El Superusuario cancela la operación.	

---

### 12. Actualizar Ítem Traducible

#### Descripción

Las expresiones presentes en el Sistema pueden ser posteriormente actualizadas.

#### Actor principal

Superusuario

#### Precondición

-

#### Criterio de validación

El Ítem Traducible seleccionado por el Superusuario se ha actualizado correctamente.

#### Curso típico de eventos

Usuario	Sistema
1. El Superusuario decide actualizar un Ítem Traducible	
	2. El Sistema muestra al usuario los Ítems Traducibles registrados hasta el momento.
3. El Superusuario selecciona el ítem que desea actualizar.	
	4. El Sistema muestra las categorías a las que pertenece ese ítem, su tipo (frase o vocabulario) y la imagen relacionada, si la tuviera.
5. El Superusuario realiza las modificaciones convenientes.	
	6. El Sistema actualiza el Ítem Traducible.
<b>Cursos alternativos</b>	
5a. El Superusuario cancela la operación.	

---

### 13. Eliminar Ítem Traducible

<b>Descripción</b>	
Las expresiones presentes en el Sistema pueden ser posteriormente eliminadas.	
<b>Actor principal</b>	Superusuario
<b>Precondición</b>	
-	
<b>Criterio de validación</b>	
El Ítem Traducible seleccionado por el Superusuario se ha eliminado del Sistema.	
<b>Curso típico de eventos</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El Superusuario decide eliminar un Ítem Traducible.	2. El Sistema muestra al usuario los Ítems Traducibles registrados hasta el momento.
3. El Superusuario selecciona el Ítem Traducible que desea eliminar.	4. El Sistema solicita la confirmación de la acción.
5. El Superusuario confirma la acción.	6. El Sistema elimina el Ítem Traducible.
<b>Cursos alternativos</b>	
5a. El Superusuario cancela la operación.	

### Crear, Actualizar y Eliminar Traducción de Ítem Traducible

<b>14. Crear Traducción de Ítem Traducible</b>	
<b>Descripción</b>	
La herramienta debe permitir traducir las expresiones a cualquiera de los idiomas disponibles.	
<b>Actor principal</b>	Usuario Registrado
<b>Precondición</b>	
-	
<b>Criterio de validación</b>	
El Usuario Registrado crea una Traducción del Ítem Traducible al idioma seleccionado.	
<b>Curso típico de eventos</b>	
<b>Usuario</b>	<b>Sistema</b>
1. El Usuario Registrado decide crear una Traducción	2. El Sistema muestra los idiomas registrados.
3. El Usuario Registrado selecciona el idioma en el que desea traducir Ítems Traducibles.	4. El Sistema muestra los Ítems Traducibles y las traducciones ya existentes en ese idioma.
5. El Usuario Registrado crea el Ítem Traducible	6. El Sistema crea una referencia a la Traducción del Ítem Traducible al idioma seleccionado.



**Cursos alternativos**

- 5a. El Usuario Registrado decide añadir audio a la Traducción.
    - 5a1. El Sistema muestra el grabador de audio.
    - 5a2. El Usuario Registrado graba el audio mediante el micrófono.
    - 5a3. El Sistema asocia el nuevo archivo de audio con la Traducción.
  - 5b. El Usuario Registrado cancela la operación.
- 

**15. Actualizar Traducción de Ítem Traducible****Descripción**

Una vez creadas las Traducciones de Ítems Traducibles podrán ser actualizadas.

**Actor principal**

Usuario Registrado

**Precondición**

-

**Criterio de validación**

El Sistema actualiza la Traducción del Ítem Traducible al idioma seleccionado por el Usuario Registrado.

**Curso típico de eventos**

Usuario	Sistema
1. El Usuario Registrado decide actualizar una Traducción	
	2. El Sistema muestra los idiomas registrados.
3. El Usuario Registrado selecciona el idioma en el que desea actualizar Traducciones de Ítems Traducibles.	
	4. El Sistema muestra los Ítems Traducibles y las traducciones ya existentes en ese idioma.
5. El Usuario Registrado actualiza la Traducción deseada	
	6. El Sistema actualiza la referencia a la Traducción del Ítem Traducible al idioma seleccionado.

**Cursos alternativos**

- 5a. El Usuario Registrado decide actualizar el audio de la Traducción.
    - 5a1. El Sistema muestra el grabador de audio.
    - 5a2. El Usuario Registrado graba el audio mediante el micrófono.
    - 5a3. El Sistema asocia el nuevo archivo de audio con la Traducción.
  - 5b. El Usuario Registrado cancela la operación.
-

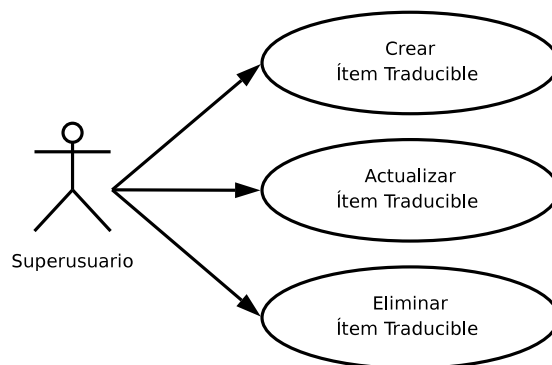


Figura 3.6: Diagrama de los casos de uso Crear, Actualizar y Eliminar Ítem Traducible.

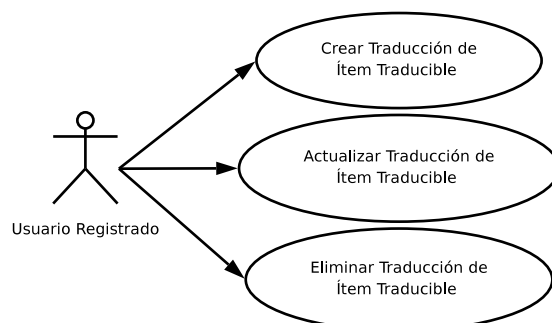


Figura 3.7: Diagrama de los casos de uso Crear, Actualizar y Eliminar Traducción de Ítem Traducible.

### 16. Eliminar Traducción de Ítem Traducible

#### Descripción

Las Traducciones presentes en el Sistema pueden ser posteriormente eliminadas.

#### Actor principal

Usuario Registrado

#### Precondición

-

#### Criterio de validación

La Traducción seleccionada por el Usuario Registrado, junto con su archivo de audio se han eliminado del Sistema.

#### Curso típico de eventos

Usuario	Sistema
1. El Superusuario decide eliminar una Traducción de un Ítem Traducible.	
	2. El Sistema muestra los idiomas registrados.
3. El Usuario Registrado selecciona el idioma en el que desea eliminar Traducciones de Ítems Traducibles.	
	4. El Sistema muestra los Ítems Traducibles y las traducciones ya existentes en ese idioma.
5. El Usuario Registrado selecciona un Ítem para eliminarlo.	

6. El Sistema elimina la referencia a la Traducción y al archivo de audio si tuviera.

#### Cursos alternativos

5a. El Usuario Registrado cancela la operación.

### Generar presentación web

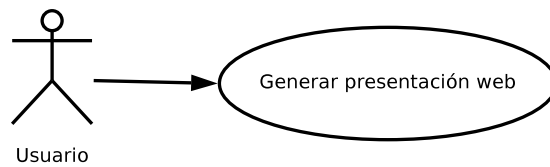


Figura 3.8: Diagrama de los casos de uso Generar presentación web.

#### 17. Generar presentación web

##### Descripción

El sistema debe permitir la visualización de expresiones de manera interactiva.

##### Actor principal

Usuario

##### Precondición

-

##### Criterio de validación

El Sistema muestra al Usuario una presentación web con los criterios seleccionados.

##### Curso típico de eventos

Usuario	Sistema
1. El Usuario decide visualizar una presentación web	
	2. El Sistema muestra dos selectores de idioma, uno para el idioma que el Usuario conoce y el otro para el idioma que el Usuario desea aprender, las categorías presentes en el sistema y los posibles tipos de ítem (todos, frase o vocabulario).
3. El Usuario selecciona las opciones convenientes.	
	4. El Sistema filtra el contenido según la selección del Usuario y muestra la presentación web que corresponde a esa selección de parámetros.

#### Cursos alternativos

4a. No existe ninguna traducción que cumpla esas características .

4a1. El sistema avisa al usuario con un mensaje de error.

### Generar ejercicios interactivos

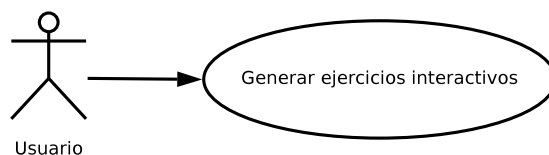


Figura 3.9: Diagrama de los casos de uso Generar ejercicios interactivos.

#### 18. *Generar ejercicios interactivos*

##### Descripción

El Sistema debe permitir la generación de ejercicios interactivos.

##### Actor principal

Usuario

##### Precondición

-

##### Criterio de validación

El Sistema muestra al Usuario una serie de ejercicios interactivos con los criterios seleccionados.

##### Curso típico de eventos

Usuario	Sistema
1. El Usuario decide visualizar ejercicios interactivos.	
	2. El Sistema muestra dos selectores de idioma, uno para el idioma que el Usuario conoce y el otro para el idioma que el Usuario desea aprender, las categorías presentes en el sistema y los posibles tipos de ítem (todos, frase o vocabulario).
3. El Usuario selecciona las opciones convenientes.	
	4. El Sistema filtra el contenido según la selección del Usuario y muestra los ejercicios interactivos que corresponden a esa selección de parámetros.

##### Cursos alternativos

4a. No existe ninguna traducción que cumpla esas características.

4a1. El sistema avisa al usuario con un mensaje de error.

### Generar PDF

#### 19. *Generar PDF*

##### Descripción

El sistema debe permitir la generación de documentos PDF.

##### Actor principal

Usuario

Precondición	
-	
Criterio de validación	
El Sistema muestra al Usuario un documento PDF generado a partir de los criterios seleccionados.	
Curso típico de eventos	
Usuario	Sistema
1. El Usuario decide generar un documento PDF	
	2. El Sistema muestra dos selectores de idioma, uno para el idioma que el Usuario conoce y el otro para el idioma que el Usuario desea aprender, las categorías presentes en el sistema y los posibles tipos de ítem (todos, frase o vocabulario).
3. El Usuario selecciona las opciones convenientes.	
	4. El Sistema filtra el contenido según la selección del Usuario y genera un documento PDF en el que se incluye el contenido de su selección acompañado de una serie de ejercicios.
Cursos alternativos	
4a. No existe ninguna traducción que cumpla esas características .	
4a1. El sistema avisa al usuario con un mensaje de error.	

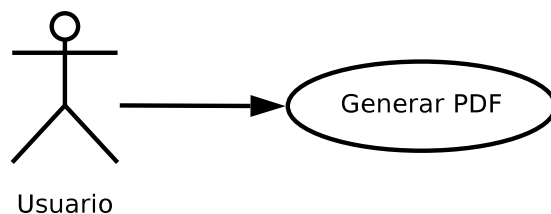


Figura 3.10: Diagrama de los casos de uso Generar PDF.

## 3.2. Modelo conceptual

El modelo conceptual es la representación de los conceptos más significativos del dominio. En el se muestran los siguientes conceptos:

**Clases de objetos** conjunto de entidades que mantienen el mismo comportamiento y las mismas propiedades.

**Asociaciones** relaciones entre clases de objetos.

**Atributos** propiedades de las clases de objetos.

**Restricciones textuales** definiciones de ciertas condiciones que debe cumplir el modelo, aquellas que no son representables mediante UML.

### 3.2.1. Diagrama de clases

La Figura 3.11 muestra el modelo conceptual completo. A continuación se introduce el diagrama de clases en partes explicando cada una de ellas.

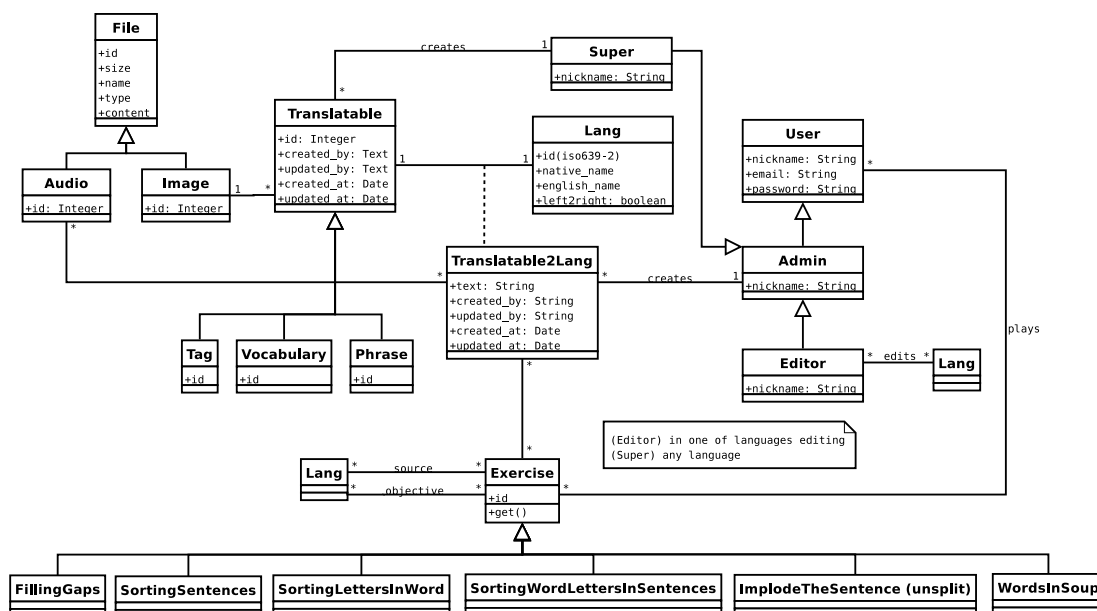


Figura 3.11: Diagrama de clases.

#### Usuarios

Se representan en el diagrama dos tipos de usuario, Super y Editor, subdivididos dentro de la subclase Admin, esta subclase aparece para englobar las operaciones comunes entre Superusuarios y Editores.

#### Idiomas

Los idiomas representados en el sistema se encapsulan dentro de la clase Lang.

#### Expresiones

Esta es la parte fundamental de la aplicación. La clase Translatable identifica a una expresión en el sistema, la clase asociativa Translatable2Lang representa una expresión en un idioma.

Obviamente una expresión debe existir en algún idioma, así pues, la clase Translatable servirá para asociar relaciones al conjunto de expresiones en diferentes idiomas que se refieren a un mismo concepto.

Las expresiones pueden tener asociadas una imagen, estas expresiones pueden ser frases, vocabulario o nombre de categoría (Tag), del diagrama se puede deducir que todas las expresiones correspondientes a un mismo concepto traducidas a diferentes idiomas compartirán la

misma imagen, en cambio es de esperar que el audio no sea el mismo para cada idioma y así se representa en el diagrama.

### Ejercicios

En el diagrama se puede observar la clase `Exercise` que representa a los ejercicios que el sistema genera. Un ejercicio es generado por un usuario y tiene asociados dos idiomas, normalmente el idioma hablado por el usuario y el idioma que desea aprender.

Los ejercicios están divididos en diferentes subclases y se diferencian principalmente por su forma de estructurar visualmente la información.

### 3.2.2. Claves externas y restricciones textuales

#### Claves externas

**User** (nickname)

**File** (id)

**Audio** (id)

**Image** (id)

**Translatable** (id)

**Translatable2Lang** (id-translatable, id-lang)

**Vocabulary** (id)

**Phrase** (id)

**Tag** (id)

**Exercise** (id)

#### Restricciones textuales

1. El nombre de usuario sólo puede contener caracteres alfanuméricos.
2. Un Usuario no puede ser Superusuario y Editor al mismo tiempo.
3. Un Editor únicamente puede crear instancias de `Translatable2Lang` en aquellos idiomas que edita.
4. Un Superusuario puede crear Superusuarios y Editores.
5. La fecha de última conexión de un Usuario debe ser mayor que su fecha de ingreso.





# Capítulo 4

## Diseño

Una vez finalizada la especificación, en la que se ha detallado el comportamiento externo de la aplicación desde el punto de vista del usuario, en esta fase se pretende definir la arquitectura de la aplicación y el tipo de plataforma en la que funcionará.

### 4.1. Decisiones de diseño

La herramienta necesita un servidor de páginas web con soporte para páginas dinámicas PHP Hypertext Pre-processor (PHP). Puesto que el lenguaje de desarrollo será PHP, es necesario un servidor de páginas web que tenga instalado el módulo de soporte para PHP.

La persistencia de los datos debe ser gestionada por MySQL. Se ha escogido MySQL por ser un gestor de bases de datos Open Source y que aporta la potencialidad adecuada a los requisitos, entre ellos la facilidad de integración a PHP.

#### 4.1.1. Plataforma física

El lenguaje de programación escogido debe ser suficientemente potente y versátil como para permitir realizar todas las funcionalidades especificadas para la aplicación. Se ha realizado la especificación utilizando la metodología de Orientación a Objetos, por tanto es necesario que el lenguaje utilizado también siga esta metodología.

Para que un lenguaje se pueda considerar orientado a objetos tiene que permitir como mínimo las características de: encapsulación, herencia, polimorfismo. PHP cumple con estas características.

#### 4.1.2. Modelo de diseño de la aplicación

Se ha elegido una arquitectura que se basa en el patrón arquitectónico en capas. Este patrón estructura la aplicación en grupos de clases, de forma que cada grupo tiene un nivel de abstracción bien diferenciado y unas tareas concretas.

Para el modelo de diseño de la aplicación se ha optado por la conjunción de dos patrones arquitectónicos.

- Patrón arquitectura en capas.

- Modelo-Vista-Controlador (MVC).

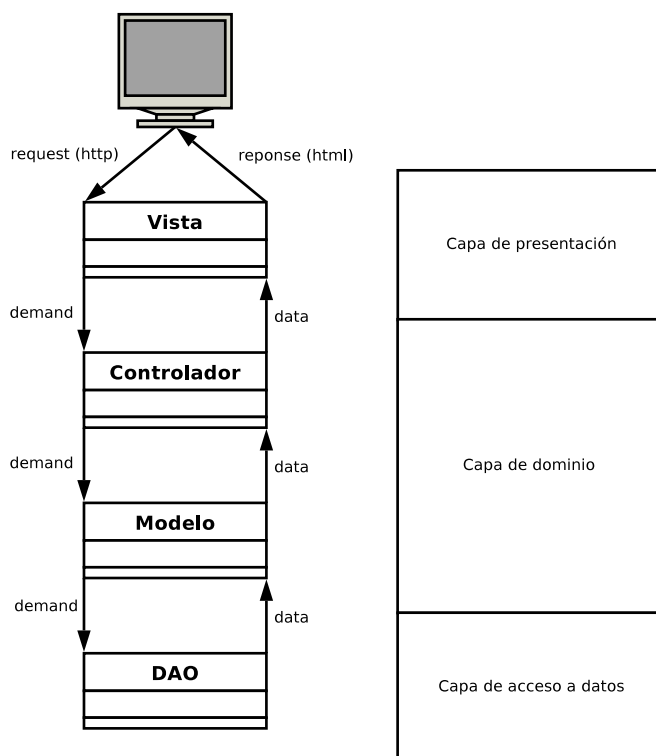


Figura 4.1: Arquitectura de la aplicación.

En concreto, las capas de la arquitectura del sistema son la capa de presentación, la capa de dominio y la capa de gestión de datos.

El MVC no pretende discriminar entre la capa de dominio y la capa de presentación pero sí pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo.

### 4.1.3. Capa de presentación

La capa de presentación es la que actúa de intermediaria entre el usuario y la capa de dominio. Los eventos generados por el usuario en la interacción con la Vista son transmitidos al Controlador correspondiente y este se encarga de comunicarse con los niveles inferiores.

En esta capa se incluyen los aspectos relacionados con la visualización final de la aplicación y la captación de eventos generados por el usuario. La capa de presentación es también la encargada de producir efectos agradables de visualización, validar la información antes de ser enviada a las siguientes capas y generar mensajes de error.

#### 4.1.4. Capa de dominio

Esta capa es la encargada de gestionar las peticiones que llegan desde la capa de presentación y procesarlas solicitando información a la capa de gestión de datos si es necesario.

La capa de dominio actúa de intermediaria entre presentación y datos, normalmente es aquella en la que se acumula la mayor carga de trabajo a la hora de implementar el sistema.

#### 4.1.5. Capa de acceso a datos

La capa de acceso a datos es responsable de la conexión con el SGBD. Las peticiones de la capa de dominio son interpretadas y procesadas por esta capa.

Para el diseño de esta capa, se han elegido los objetos DAO, los cuales están asociados a las tablas presentes en el SGBD. Los objetos DAO serán expuestos con más detalle más adelante.

## 4.2. Diseño de la arquitectura del sistema

### 4.2.1. Diseño de la capa de presentación

El diseño de pantallas debe permitir al usuario la consecución de todos los casos de uso y tener en cuenta los requisitos funcionales. Para el diseño de pantallas se ha utilizado la versión gratuita de Balsamiq Mockups [7], un editor de pantallas prototipo, que pese a no ser open source, no tiene coste alguno por tratarse de una versión gratuita.

A continuación se muestra el diseño inicial de algunas de las pantallas más representativas de la aplicación.

#### **Pantalla: Presentación web**

La Figura 4.2 corresponde al diseño de la pantalla que mostrará las presentaciones web.

1. Menú principal.
2. Selector de idioma (El idioma que el usuario habla).
3. Selector de idioma (El idioma que el usuario aprende).
4. Selector de sección. (Puede seleccionarse Presentación web, ejercicios interactivos y ejercicios en PDF)
5. Selector de categorías. (Filtra el contenido que se muestra según la categoría)
6. Selector de imagen. (Muestra en pequeño las imágenes que corresponden a la selección)
7. Imagen que representa el ítem actual.
8. Ítem en el idioma que el usuario ha seleccionado que habla.
9. Ítem en el idioma que el usuario ha seleccionado que aprende.
10. Botón para reproducir el audio correspondiente al ítem mostrado.

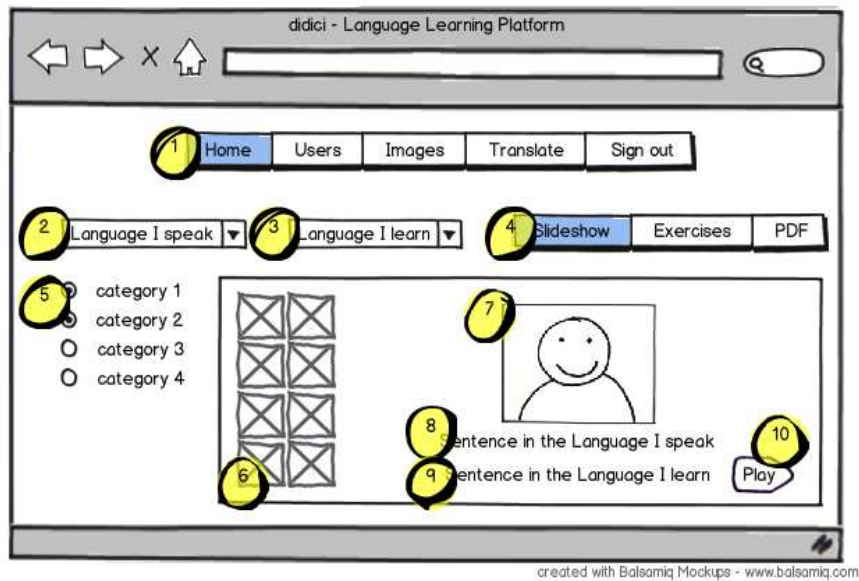


Figura 4.2: Diseño de la pantalla que muestra presentaciones.

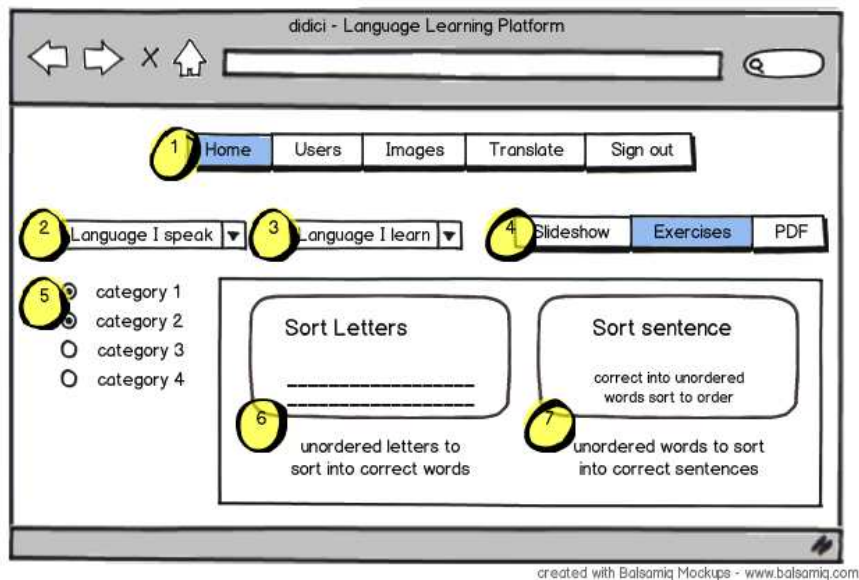


Figura 4.3: Diseño de la pantalla que muestra los ejercicios. Dos tipos de ejercicio.

### Pantalla: Ejercicios

La Figura 4.3 y la Figura 4.4 corresponden al diseño de la pantalla que muestra los ejercicios. En este caso, se muestran cuatro tipos de ejercicio.

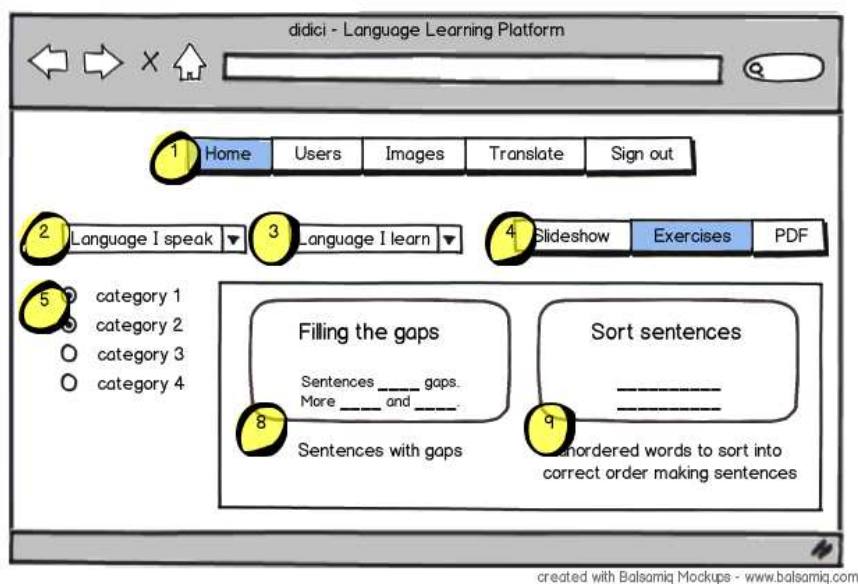


Figura 4.4: Diseño de la pantalla que muestra los ejercicios. Otros dos tipos de ejercicio.

En la aplicación se mostrará un listado con los ejercicios que resulten de la selección del usuario. Se mostrará un ejercicio cada vez y los ejercicios serán de los cuatro tipos que se especifican a continuación.

1. Menú principal.
2. Selector de idioma (El idioma que el usuario habla).
3. Selector de idioma (El idioma que el usuario aprende).
4. Selector de sección. (Puede seleccionarse Presentación web, ejercicios interactivos y ejercicios en PDF)
5. Selector de categorías. (Filtra el contenido que se muestra según la categoría)
6. *Sort Letters*: Las frases en las que el usuario aprende aparecen con los caracteres desordenados dentro de cada palabra.
7. *Sort Sentence*: Aparece una frase con las palabras desordenadas.
8. *Filling the gaps*: Aparecen diferentes frases en las que hay huecos. Abajo aparecen desordenadas las palabras que corresponden a los huecos.
9. *Sort sentences*: Aparecen diferentes frases en el idioma hablado que tienen que ser compuestas en el idioma aprendido con las palabras desordenadas que aparecen más abajo.

#### Pantalla: Nuevo ítem

La Figura 4.5 muestra el diseño de la pantalla que servirá para el registro de un nuevo ítem. El nuevo ítem se registra en el idioma inglés, puede ser asociado a diferentes categorías y asociado a una imagen, además podrá ser categorizado como frase o vocabulario.

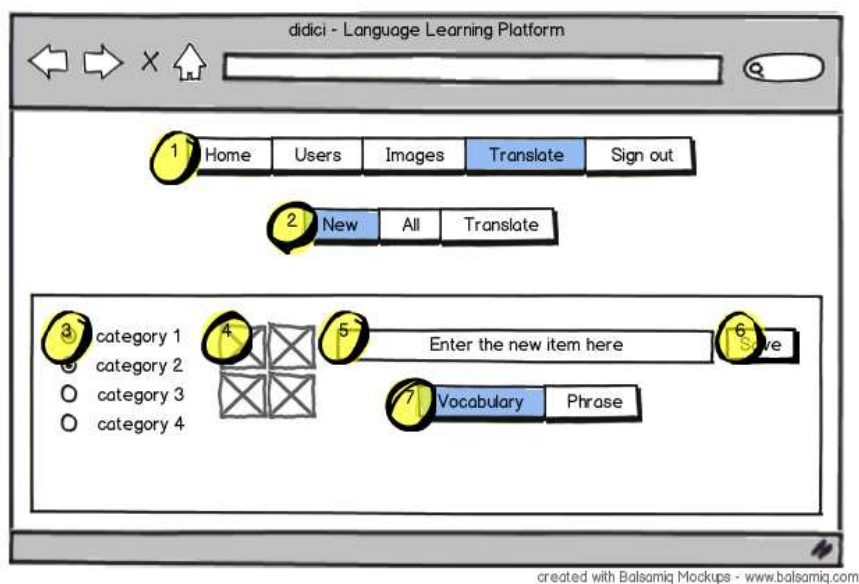


Figura 4.5: Diseño de la pantalla para registrar un nuevo ítem.

1. Menú principal.
2. Menú de sección.
3. Selector de categoría (Las categorías que se asociarán al nuevo ítem).
4. Selector de imagen. (Imagen que será asociada con el nuevo ítem)
5. Campo de texto para introducir el texto correspondiente al nuevo ítem.
6. Guarda el nuevo ítem.
7. Categoriza el ítem como vocabulario o frase.

#### **Pantalla: Zona de traducción**

La Figura 4.6 muestra la zona de traducción de la aplicación en la que aparecen los idiomas seleccionables dentro del sistema y las frases que pueden ser traducidas.

Las traducciones se realizan siempre desde el idioma inglés hacia cualquier otro idioma presente en el sistema. Es en la zona de traducción donde es posible grabar audio asociado a cada ítem en cada idioma.

1. Menú principal.
2. Menú de sección.
3. Selector de idioma (El idioma en que se van a traducir frases).
4. Frase en el idioma inglés.

5. Campo de texto para introducir la traducción correspondiente.
6. Grabador de audio.
7. Guarda los cambios realizados en la frase.

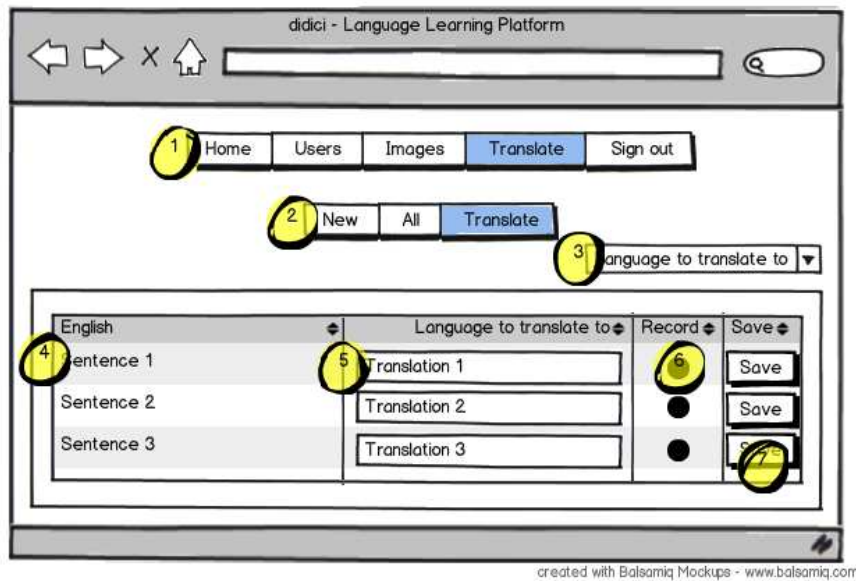


Figura 4.6: Diseño de la pantalla que muestra la zona de traducción.

#### 4.2.2. Diseño de la capa de dominio

La capa de dominio, intermedia entre la presentación y el acceso a los datos es la encargada de gestionar las peticiones que proceden de la vista.

##### Controlador

El patrón arquitectónico Controlador de Fachada permite encapsular todas las operaciones sobre una misma clase de objetos dentro de un mismo controlador, así se mantiene la escalabilidad del código.

Las operaciones relacionadas con Usuario se encapsularán dentro de UserController y todo lo relacionado con expresiones (creación, consulta, traducción, etc...) estará encapsulado en el TranslatableController, y así con cada una de las Entidades representativas, en la Figura 4.7 se muestran todos los controladores del sistema.



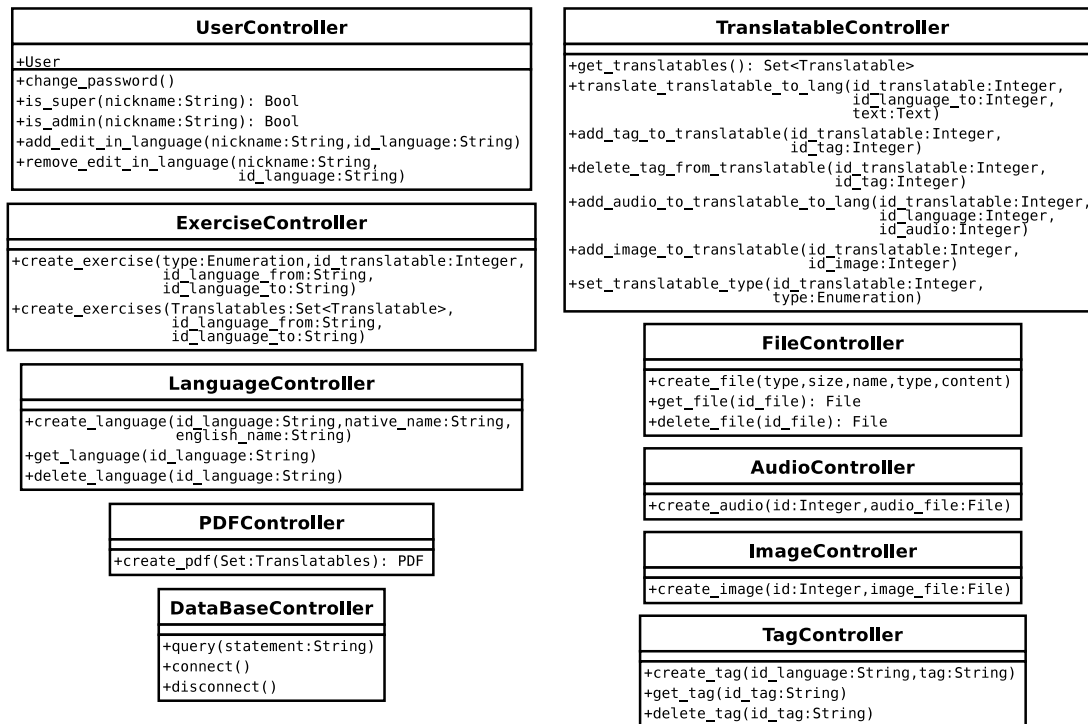


Figura 4.7: Diseño de los controladores del sistema.

### 4.2.3. Diseño de la capa de acceso a datos

Se utilizarán objetos DAO para la conexión con la base de datos. Existirá un objeto DAO para cada tabla de la base de datos, los atributos de este objeto corresponderán a los campos de esa tabla. La Figura 4.8 muestra las operaciones comunes para los objetos DAO diseñados para la construcción de la herramienta.

Cada tabla estará asociada a un objeto DAO, estas son las operaciones que deberán ser implementadas:

**DAOCreate** creación de registros en la base de datos.

**DAOLoad** consulta de registros en la base de datos y carga de atributos del objeto.

**DAOUpdate** actualización de registros en la base de datos.

**DAODelete** elimina registros de la base de datos.

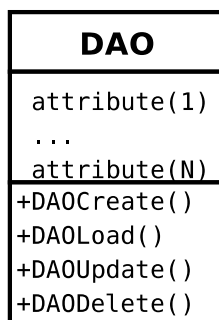


Figura 4.8: Objeto DAO.

Cuadro 4.1: Estructura de la tabla Audio

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id</i>	int(11)	No

Cuadro 4.2: Estructura de la tabla Editing

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id_editor</i>	varchar(128)	No
<i>id_language</i>	varchar(3)	No

Cuadro 4.3: Estructura de la tabla Editor

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id</i>	varchar(64)	No

Cuadro 4.4: Estructura de la tabla File

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id</i>	int(11)	No
name	varchar(128)	No
type	varchar(64)	No
size	int(11)	No
content	longblob	No

Cuadro 4.5: Estructura de la tabla IText

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id</i>	varchar(64)	No

Cuadro 4.6: Estructura de la tabla IText2Lang

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id_itext</i>	varchar(64)	No
<i>id_language</i>	varchar(3)	No
itext	text	No

Cuadro 4.7: Estructura de la tabla Language

Campo	Tipo	Nulo
<i>id</i>	varchar(3)	No
native_name	varchar(64)	No
english_name	varchar(64)	No
left2right	tinyint(4)	No

Cuadro 4.8: Estructura de la tabla Phrase y la tabla Vocabulary

Campo	Tipo	Nulo
<i>id</i>	int(11)	No

Cuadro 4.9: Estructura de la tabla Resource

Campo	Tipo	Nulo
<i>id</i>	int(11)	No

Cuadro 4.10: Estructura de la tabla Super

Campo	Tipo	Nulo
<i>id</i>	varchar(64)	No

Cuadro 4.11: Estructura de la tabla Tag

Campo	Tipo	Nulo
<i>id</i>	int(11)	No

Cuadro 4.12: Estructura de la tabla Tag2Lang

Campo	Tipo	Nulo
<i>id_tag</i>	int(11)	No
<i>id_language</i>	varchar(3)	No
tag	text	No

Cuadro 4.13: Estructura de la tabla Translatable

Campo	Tipo	Nulo
<i>id</i>	int(11)	No
created_by	varchar(64)	No
created_at	datetime	No
updated_by	varchar(64)	No
updated_at	datetime	No

Cuadro 4.14: Estructura de la tabla Translatable2Lang

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id_translatable</i>	int(11)	No
<i>id_language</i>	varchar(3)	No
text	text	No
created_by	varchar(64)	No
created_at	datetime	No
updated_by	varchar(64)	No
updated_at	datetime	No

Cuadro 4.15: Estructura de la tabla Translatable2Lang2Audio

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id_translatable</i>	int(11)	No
<i>id_language</i>	varchar(3)	No
id_audio_file	int(11)	No
created_by	varchar(64)	No
created_at	datetime	No
updated_by	varchar(64)	No
updated_at	datetime	No

Cuadro 4.16: Estructura de la tabla Translatable2Resource

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id_translatable</i>	int(11)	No
<i>id_resource</i>	int(11)	No

Cuadro 4.17: Estructura de la tabla Translatable2Tag

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
<i>id_translatable</i>	int(11)	No
<i>id_tag</i>	int(11)	No

Cuadro 4.18: Estructura de la tabla User

<b>Campo</b>	<b>Tipo</b>	<b>Nulo</b>
email	varchar(128)	No
<i>name</i>	varchar(64)	No
password	varchar(64)	No
registered	datetime	No
last_login	datetime	No

# Capítulo 5

## Implementación

El objetivo de esta etapa es implementar el diseño obtenido en la etapa anterior con una tecnología concreta. Esto permitirá poder ejecutarlo y utilizarlo.

En el capítulo anterior se decidió utilizar PHP como lenguaje principal de programación, en este capítulo se expondrán algunas de las características de este y otros lenguajes de programación utilizados en el desarrollo de la plataforma así como la estrategia de desarrollo utilizada y otra información de interés.

### 5.1. Estrategia de desarrollo

La estrategia de desarrollo escogida es el *Desarrollo iterativo e incremental* el cual empieza con una planificación inicial y acaba con el despliegue final, entre ambas fases se repiten de forma cíclica las fases que se observan en la Figura 5.1.

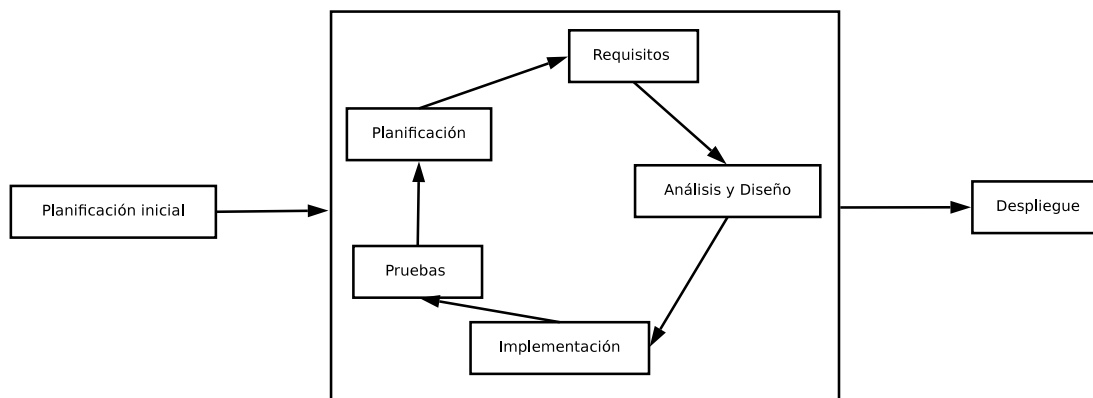


Figura 5.1: Ciclo de vida del desarrollo iterativo e incremental

### 5.2. Tecnologías utilizadas (LAMP)

El acrónimo 'LAMP' se refiere a un conjunto de subsistemas de software necesarios para alcanzar una solución global, en este caso configurar sitios web o servidores dinámicos con un

esfuerzo reducido.

En las tecnologías LAMP esto se consigue mediante la unión de las siguientes tecnologías:

- Sistema Operativo **Linux**
- Servidor web **Apache**
- Gestor de Base de Datos **MySQL**
- **PHP**

Para la realización de este proyecto se han utilizado las tecnologías LAMP.

### 5.3. Entorno de desarrollo

**Sistema Operativo** Debian 5.07 Kernel Linux 2.6.26-2-686

**IDE** Eclipse para desarrolladores de PHP

**Servidor web** Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny9

### 5.4. Lenguajes de programación utilizados

A continuación se definen algunas de las características de los lenguajes de programación utilizados para la implementación de la plataforma, junto con algunas de las porciones del código de la aplicación.

#### 5.4.1. HTML

*HTML*, siglas de HyperText Markup Language («lenguaje de marcado de hipertexto»), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. *HTML* se escribe en forma de «etiquetas», rodeadas por corchetes angulares (< >).

*HTML* también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo *JavaScript*), el cual puede afectar el comportamiento de navegadores web y otros procesadores de *HTML*.

El código *HTML* presente en la aplicación es generado mediante *PHP*. En el siguiente ejemplo de código se muestra un archivo plano de *HTML* que será interpretado por *PHP*, que sustituirá los elementos entre *arrobas* por el código *HTML* correspondiente en función del estado de la aplicación.

```

0  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML1.0 Transitional//EN" "http://
    www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title> @@PAGE_TITLE@@ </title>
5
    <link rel="stylesheet" href="/path/to/files.css" type="text/css" />
    <script src="/path/to/files.js" type="text/javascript"></script>
  </head>
  <body>
10  <div id="topbar"> @@TOPBAR@@ </div>
    <div id="page">
      @@HEADER@@
      <div id="content"> @@MAIN@@ </div>
      @@FOOTER@@
15  </div>
  </body>
</html>

```

### 5.4.2. CSS

Las hojas de estilo en cascada (Cascading Style Sheets, *CSS*) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en *HTML* o *XML* (y por extensión en *XHTML*). El *W3C* (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servir estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de *CSS* es separar la estructura de un documento de su presentación.

Las hojas de estilo están presentes en la aplicación en varios archivos, algunos han sido generados mediante herramientas existentes, como es el caso de *jQueryUI* que además de proporcionar elementos gráficos que mejoran la interacción incluyen sus propias hojas de estilo y sus etiquetas específicas.

```

0  /*****
   * Ejemplo: css para topbar
   *****/
   #topbar { text-align: right; margin: 0; padding: 0; font-size: 0.9em;
           background-color: black; }
5  #topbar-inner { padding: 0.3em; margin: 0 auto; width: 88%; }
   #login-form { padding: 0; margin: 0; }
   #login-form ul { list-style: none; padding: 0; margin: 0; }
   #login-form li { padding: 0; margin: 0; }
   #login-form input { width: 99%; margin-bottom: 0.5em; }
   #login-form #login-submit { width: auto; float: right; font-size: 0.8
           em; }

```

### 5.4.3. PHP

*PHP* es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando diferentes bibliotecas.

La aplicación está mayormente desarrollada en *PHP*, este lenguaje implementa los controladores y el modelo, que se conecta al *SGBD* mediante las librerías específicas en este lenguaje. La vista está implementada igualmente en este lenguaje que es el encargado de acabar generando el código *HTML* accesible para los usuarios.

```

0  <?
   include_once 'lib/common/config/autoloading.php';

   /*****
5  * Depending on the user role we show the corresponding page
   *
   *****/

   // SessionController allows us to ask which is the user role
10  $SessionController = SessionController::getInstance();

   if ($SessionController->isGuest()) echo new Templater(BASEPATH."/lib/app
       /view/main/page.tpl", new guest());
   else echo new Templater(BASEPATH."/lib/app/view/main/page.tpl", new
       home_page());

15  ?>

```

#### 5.4.4. JavaScript

*JavaScript* es un lenguaje de programación interpretado ejecutado principalmente en el lado del cliente, normalmente está implementado en la parte del navegador web y se utiliza principalmente para mejorar la experiencia del usuario. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

En esta aplicación se ha utilizado *jQuery*, una librería de *JavaScript* que permite un desarrollo más ágil reduciendo el código necesario para realizar las mismas funciones, además de contar con numerosos plugins con licencias *OpenSource*.

```

0  <script>
   /* Using JQuery */
   $(function(){
       reload();

5   $('#selected_tags_by_user_option').click(function(){ reload(); });

       function reload()
       {
10          $('#selected_tags_by_user_list').empty();
           $('#selected_tags_by_user_option:selected').each(function(){
               $('#selected_tags_by_user_list').prepend("<li>"+$(this).text()+
                   "</li>");
           });
       }
   });
15 </script>

```

#### 5.4.5. Java

*Java* es un lenguaje de programación orientado a objetos, desarrollado por *Sun Microsystems* a principios de los años 90.



En este proyecto, el grabador de audio está integrado en el portal web por medio de un signed applet.

```
0 private void captureAudio(){
  try{
    //Get things set up for capture
    audioFormat = getAudioFormat();
    DataLine.Info dataLineInfo =
5         new DataLine.Info(
            TargetDataLine.class,
            audioFormat);
    targetDataLine = (TargetDataLine)
10        AudioSystem.getLine(dataLineInfo);

    //Create a thread to capture the microphone
    // data into an audio file and start the
    // thread running. It will run until the
    // Stop button is clicked. This method
15    // will return after starting the thread.
    new CaptureThread().start();
  }catch (Exception e) {
    e.printStackTrace();
    System.exit(0);
20  } //end catch
  } //end captureAudio method
```

## 5.5. Integración de plugins y otras herramientas

Para la realización de la plataforma se han utilizado diferentes plugins y otras clases de otros desarrolladores. La integración de otros programas ya existentes para la implementación de la plataforma permite reducir el tiempo de desarrollo, si no se dispusiera de estos plugins algunas de las funcionalidades deberían ser descartadas precisamente por falta de tiempo.

### 5.5.1. TFPDF

Esta clase se distribuye bajo licencia *LGPL*, desarrollada por Ian Back. Es una versión modificada de *FPDF*, añadiendo soporte *UTF-8*. Además, sólo incluye las partes necesarias de las fuentes usadas en cada documento, haciendo que el tamaño de archivo sea más pequeño. [8]

### 5.5.2. Galleriffic

Galleriffic es un plugin para *jQuery* para la renderización de galerías de fotos. Gracias a este plugin, la carga de imágenes se produce de forma asíncrona, lo que permite una mejor experiencia de usuario. [9]

### 5.5.3. Flowplayer

Flowplayer es un complemento reproductor de contenido multimedia para páginas web. El programa es software libre licenciado bajo la *GPL 3+*. No obstante, otras versiones particulares son bajo copyright. El programa funciona mediante un complemento *JavaScript* que puede ejecutar *Flash Video*. [10]

#### 5.5.4. `upload.class.php`

Este script *PHP* carga imágenes en el servidor y las manipula de forma sencilla. Es perfecto para generar thumbnails (imágenes pequeñas de muestra) o crear galerías de fotos. Convierte, redimensiona y trabaja con imágenes de diferentes extensiones, aplica efectos, añade etiquetas, marcas de agua y reflexiones y otras funcionalidades de edición de imágenes. Se distribuye bajo la licencia *GPL Version 2*. [11]

#### 5.5.5. `datatables`

DataTables es un plugin para *jQuery* dedicado a la renderización de tablas para mejorar la interacción con los elementos, permitiendo búsquedas, ordenaciones y carga asíncrona. [12]

# Capítulo 6

## Pruebas

### 6.1. Prueba modular

La prueba modular consiste en la creación independiente de cada una de las instancias que se muestran a continuación y comprobar que el sistema devuelve la respuesta esperada.

- Upload de una Imagen.
- Creación de Ítems Traducibles.
- Previsualización de Ítems Traducibles.
- Traducción de Ítems.
- Grabación de audio.
- Generación de ejercicios.
- Generación de documentos PDF.

### 6.2. Prueba en desarrollo

Con esta prueba se determina el funcionamiento correcto de las diferentes partes del sistema integradas entre sí. Para la realización de estas pruebas se han creado diferentes usuarios y se han reproducido las siguientes situaciones:

- Upload de Imágenes por diferentes administradores.
- Creación de Ítems Traducibles por diferentes administradores.
- Previsualización de Ítems Traducibles.
- Traducción de Ítems por editores y administradores.
- Grabación de audio por editores y administradores.
- Generación de ejercicios en diferentes idiomas.
- Interacción con los ejercicios en diferentes idiomas.
- Generación de documentos PDF en diferentes idiomas.

### 6.3. Prueba global de la aplicación

Después de integrar las diferentes funcionalidades en un sistema único, éste se ejecuta en un entorno real y con la colaboración de diferentes usuarios ajenos al desarrollo de la aplicación.

El sistema se pone en funcionamiento a la espera de que los usuarios reporten cualquier anomalía en el funcionamiento del aplicativo.

Tras un periodo de mejora y refinamiento se puede afirmar que la aplicación funciona correctamente.

# Capítulo 7

## Conclusión

Después de completar todas las etapas del proyecto, es necesario hacer una valoración de las mismas así como conocer las posibles líneas de ampliabilidad que puedan desarrollarse.

### 7.1. Estudio de la planificación

Se planteó inicialmente un diseño iterativo ascendente en el que se definieron tres iteraciones.

**Iteración 1** El objetivo definido en esta iteración fue la implementación del núcleo del sistema.

**Iteración 2** En esta iteración se planeó la implementación del grabador de audio y los primeros prototipos de ejercicios.

**Iteración 3** En esta última iteración el objetivo era el ensamblaje final de todas las tecnologías y la obtención de una versión funcional.

1. La primera iteración consistió en la creación del esqueleto del sistema (conexiones con el SGBD, objetos DAO, creación del modelo y algunos controladores). Se realizaron las primeras pruebas de diseño, se estructuraron los contenidos y se realizaron las primeras pruebas multi-lenguaje para la correcta visualización de los caracteres en diferentes idiomas.

Se planteó un diseño inicial de la interfaz y se incorporaron algunas funcionalidades como la creación de usuarios. Esta primera iteración permitió la adquisición de conocimientos importantes a nivel de tecnologías utilizadas en la implementación.

2. La segunda iteración aprovechó el esqueleto de la primera e introdujo la grabación y reproducción de audio. En esta segunda etapa, el sistema no contaba con parte de administración de contenidos pero sí con parte pública. Los contenidos se estructuraban en cursos con lecciones y cada lección tenía una serie de ítems, cada uno de ellos con imagen, audio y la frase en los dos idiomas de ese curso. Además se realizaron los primeros prototipos de ejercicios pero no se integraron dentro de las funcionalidades del sistema.

En el momento de realizar la planificación de la siguiente iteración, ante el riesgo de no cumplir con los objetivos por falta de tiempo, se decide reducir las funcionalidades iniciales del sistema y modificar el planteamiento inicial (ver Apéndice A).

- En la tercera iteración los contenidos pasan a ser ítems en un idioma que pueden ser traducidos a cualquier otro idioma presente en el sistema. Estos ítems pueden ser categorizados y el usuario puede seleccionar ítems en función de la categoría.

Los prototipos de las primeras iteraciones pasan a ser módulos de la aplicación que funcionan de forma coordinada. La parte de administración de contenidos y la parte pública se acaban de concretar en esta iteración, se añade el grabador obtenido en la iteración anterior y se integra la generación de ejercicios dentro de las funcionalidades del sistema.

La flexibilidad y usabilidad de esta tercera iteración cumple con los objetivos fijados y se da la implementación por terminada.

### Planificación inicial vs. planificación real

La planificación inicial contaba con una serie de tareas definidas que debían realizarse para alcanzar los objetivos. Estas tareas, definidas en el punto 1.5 de este documento tenían un tiempo estimado para su consecución que varió con el tiempo.

A continuación se muestra una tabla comparativa entre las horas estimadas y las horas reales empleadas.

Tarea	Horas estimadas	Horas reales
Descripción del proyecto y objetivos	40	30
Análisis de requisitos	80	60
Especificación	160	180
Diseño	200	180
Aprendizaje de las tecnologías	100	120
Implementación	240	300
Pruebas	20	40
Documentación	60	80
<b>Total</b>	<b>900</b>	<b>990</b>

Como se muestra en la tabla, finalmente se emplearon más horas de las previamente estimadas para la implementación. Esto fue debido a las iteraciones realizadas y al hecho de que la integración de tecnologías resultó dificultosa en cuanto a la grabación de sonido. No obstante, estas iteraciones eran totalmente necesarias y gracias a ellas el resultado final del proyecto es el que es.

Si el proyecto volviera a realizarse desde cero para un cliente real, existirían algunas decisiones que se tomarían de forma diferente. Por ejemplo, en lugar de exceder el número de horas estimadas para la grabación de audio y su posterior reproducción, se hubiera optado por algún producto de pago, que permitiera esa funcionalidad y ahorrar el coste en la investigación de la tecnología, se utilizaría algún tipo de estándar para la representación de la información multi-idioma y su mejor interpretación por los motores de búsqueda, la generación de ejercicios se realizaría de forma externa mediante un plugin para su mayor reutilización y un sin fin de decisiones fácilmente cambiables a posteriori.

## 7.2. Estudio económico

### 7.2.1. Coste de desarrollo

<i>Recursos software</i>	
<b>Sistema operativo</b>	Debian 5.07 Kernel Linux 2.6.26-2-686
<b>Entorno de desarrollo</b>	IDE Eclipse para desarrolladores de PHP
<b>Servidor web</b>	Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny9
<b>Manipulación gráfica</b>	GIMP GNU image Manipulation Program
<b>Editor de diagramas</b>	Dia Diagram Editor
<i>Recursos hardware</i>	
<b>Estación de trabajo</b>	AMD Athlon X2 250 Processor 4GiB memory

Todos los recursos software son herramientas open source, como se indicaba en el punto 2.2.4 y tienen coste 0. El precio del hardware dependerá de la calidad del mismo. En nuestro caso, ronda el precio aproximado de unos 600 euros. Suponiendo que el ordenador será válido para la realización de otros proyectos, la inversión inicial del hardware se amortizará a medida que se realicen nuevos proyectos.

### 7.2.2. Coste de implantación

El mínimo coste de implantación es el precio de un ordenador que funcione de servidor si se quiere trabajar en un red local o en su defecto, si lo que se quiere es tener la plataforma accesible desde cualquier parte del mundo, lo ideal sería la contratación de un servidor lo suficientemente potente para recibir las transacciones de la aplicación.

Existen diferentes tipos de servidores, la elección de uno u otro variará en función del tráfico que genere el portal. Siendo los precios aproximados de:

- Servidor compartido: 8 euros mensuales.
- Servidor virtual privado: 100 euros mensuales + entre 10 euros/mes y 150 euros/mes dependiendo de los Megabytes de memoria contratada.

### 7.2.3. Recursos humanos

En este apartado se pretende cuantificar el coste de la mano de obra necesaria para la realización del proyecto. Se contabilizan las horas dedicadas a la realización del proyecto sin tener en cuenta las horas de formación.

Se distinguen 4 roles:

**Jefe de proyecto** Se encarga de la gestión del proyecto: planificación, viabilidad, estudio económico y asignación y priorización de tareas.

**Analista** Se le asignan las tareas de captura de requisitos, análisis y diseño.

**Programador** Se encarga de la implementación y las pruebas.

**Diseñador gráfico** Se le asignan las tareas del diseño visual del aplicativo.

A continuación se muestra una tabla con los precios aproximados por hora de cada uno de los recursos descritos. Estos precios pueden variar en función de diferentes variables como la calidad de los servicios recibidos, la formación de los profesionales contratados o la demanda del mercado, entre otras.

Recurso	Precio (€/hora)
Jefe de proyecto	40
Analista	30
Programador	20
Diseñador gráfico	15

Se muestra ahora una estimación del coste de los recursos humanos necesarios para la realización del proyecto en el supuesto de que se realizara para un cliente. No se tienen en cuenta las horas de formación en las tecnologías ni el diseño gráfico.

Tarea	Horas	Recurso	Coste (€)
Descripción del proyecto y objetivos	30	Analista	900
Análisis de requisitos	60	Analista	1800
Especificación	120	Analista	3600
Diseño	120	Analista	3600
Implementación	240	Programador	4800
Pruebas	40	Programador	800
Gestión del proyecto	80	Jefe de proyecto	3200
<b>Total</b>	<b>690h</b>		<b>18700 €</b>

### 7.3. Objetivos conseguidos

El objetivo del proyecto era realizar una aplicación que permitiera la creación de ejercicios interactivos a partir de frases previamente introducidas en el sistema y traducidas en diferentes idiomas. Las funcionalidades del sistema son las que se propusieron inicialmente en el punto 2.2.6. El sistema cumple con los requisitos funcionales y no funcionales especificados y permite:

- La creación de usuarios de diferentes tipos. Superusuario, editor e invitado.
- Añadir nuevas imágenes en el sistema.
- Añadir nuevos ítems y traducirlos a diferentes idiomas.
- Añadir categorías y traducirlas a diferentes idiomas.
- Grabar audio para cada uno de los ítems del sistema en cada idioma.
- Categorizar los ítems presentes en el sistema, asignarles una imagen y un registro de audio.
- Generar una presentación web con los ítems correspondientes a la selección por parte del usuario.
- Generar ejercicios interactivos a partir de la selección de idiomas y de categorías realizada por el usuario.
- La generación de ejercicios en formato PDF, para su realización sobre papel.



Uno de los objetivos iniciales que no se ha alcanzado es la internacionalización de la plataforma. El exceso de tiempo dedicado al ensamblaje de tecnologías hizo retrasar esta tarea hasta el punto de no alcanzarse finalmente. No obstante, esta tarea no es bloqueante para la consecución total de los objetivos principales pero si es deseable, así pues se ha trasladado a la primera tarea en orden de prioridad de las líneas de futuro, que se muestran a continuación.

## 7.4. Líneas de futuro

**i18n** Como se menciona en el punto anterior, la primera de las tareas a realizar en el futuro es la internacionalización de la plataforma, es decir, que la interfaz se presente en el idioma escogido por el usuario. Para esto, se podrían utilizar diferentes técnicas de internacionalización. Cualquiera de las siguientes opciones sería válida para conseguir la internacionalización de la plataforma:

- Almacenar las traducciones en un fichero XML para cada idioma y sustituirlas en función del idioma seleccionado.
- Se podrían almacenar las traducciones de la interfaz en formato JSON y mediante javascript refrescar los contenidos susceptibles de ser traducidos.
- Otra técnica es la utilización de gettext, una herramienta que integrada con PHP genera archivos binarios para las traducciones que se sustituyen directamente en tiempo de ejecución.

**Volcado masivo de frases** Otra tarea sería la implementación de una funcionalidad que permitiera el volcado masivo de frases y traducciones, esto conllevaría la búsqueda, preparación y posterior utilización de algún conjunto de frases y traducciones existente, o la creación de uno propio por parte de editores o administradores y que reduciría el tiempo que estos emplean para registrar nuevos ítems.

**Mejora de la velocidad** Para mejorar la velocidad de la plataforma se podría implementar un sistema de caches que permitiera el acceso a las frases de forma más rápida.

Se podrían generar uno o varios archivos planos, XML o JSON con la información accesible de las frases registradas en el sistema. Estos archivos se regenerarían periódicamente mediante un proceso automático con la información más reciente de la base de datos.

**Dos ramas diferenciadas** Existen varias líneas de ampliabilidad diferenciadas dependiendo de la finalidad que se le quiera dar al proyecto.

En el caso de tener **finés educativos**, las líneas de futuro pasarían por la creación de más tipos de ejercicios y funcionalidades:

- Introducción de los roles profesor y alumno.
- Seguimiento del rendimiento del alumno.
- Intercambio de ejercicios entre profesor y alumno.
- Creación de tests de evaluación puntuables.

- Creación automática de dictados.
- Posibilidad de añadir transcripciones fonéticas.

En el caso de tener **finés comerciales**, las líneas de futuro por las que debería pasar la aplicación serían:

- Cambiar la introducción de contenidos a un formato de red social, donde mediante sistemas de confianza se consiga que el contenido sea generado y corregido por los propios usuarios.
- Diseñar e implementar las plataformas para los diferentes dispositivos móviles.
- Creación de nuevos tipos de ejercicios.
- Buscar nuevos formatos de competitividad (positiva) entre usuarios. Esto es, diseñar competiciones que motiven a los usuarios a auto-evaluarse.
- En el caso de obtener un número suficiente de visitas, obtención de ingresos mediante publicidad específica o venta de espacio para contenidos personalizados.

## 7.5. Conocimientos obtenidos

La realización del proyecto me ha valido para formarme en diferentes tecnologías y asimilar mejor los conceptos adquiridos durante la realización de la carrera en el ámbito de la Ingeniería del Software, realizando especificaciones, análisis de requisitos, planificaciones y valoraciones económicas.

Los lenguajes de programación utilizados (HTML, PHP, CSS, Javascript, Java) y la integración de diferentes plugins me han acercado a la realización de proyectos web en entornos reales, además, administrando el servidor web y configurando entornos de prueba he consolidado mis conocimientos sobre administración de sistemas Linux.

## 7.6. Valoración personal

Llevar a cabo este proyecto ha supuesto una búsqueda constante de información y formación en las tecnologías utilizadas y me ha servido para incrementar mi motivación y mi capacidad de resolver problemas buscando las mejores soluciones. La realización de este proyecto me ha aportado experiencia en el uso de nuevas tecnologías y en la gestión de proyectos.

Gracias al esfuerzo durante todo este tiempo se puede observar en mí un crecimiento tanto académico como personal lo que me supone estar muy orgulloso y satisfecho del resultado obtenido.

# Apéndice A

## Análisis de las iteraciones

**Evolución del diagrama de clases** Con cada iteración el diagrama de clases evoluciona para cumplir con mayor grado los objetivos. La Figura A.1 muestra el diagrama de clases de la primera y la segunda iteración.

Se puede observar que:

- El sistema se organiza en cursos y estos en lecciones.
- Las lecciones guardan información sobre el carácter sintáctico de cada palabra en la oración.
- Existen diferentes tipos de ejercicio pero su generación no es automatizada.

Tras analizar el diagrama, el tiempo empleado y hacer una previsión futura se llega a las siguientes conclusiones:

- La primera y la segunda iteración han sido demasiado optimistas en cuanto a las funcionalidades de la plataforma.
- El simple hecho de guardar información sobre cada una de las palabras hace que el tiempo de implementación se dispare.
- Los cursos 'fijos' de un idioma a otro no maximizan las posibilidades de generar ejercicios, e implementar la traducción de cursos incrementaría mucho el tiempo de implementación.
- Algunos de los tipos de ejercicios inicialmente propuestos son difícilmente implementables de forma automática, lo que supondría un exceso en el tiempo de implementación de las pantallas de configuración de estos ejercicios.

Finalmente y viendo que si el proyecto sigue por este camino el tiempo de implementación se disparará, se decide pasar a una tercera iteración donde los cursos y las lecciones desaparecen, ahora se pueden categorizar las frases y una misma frase puede ser traducida a cualquier idioma de la aplicación.

No se guardará información sintáctica de las palabras en la oración y únicamente se implementarán ejercicios generados de forma automática, descartando la posibilidad de que administradores y editores diseñen ejercicios.

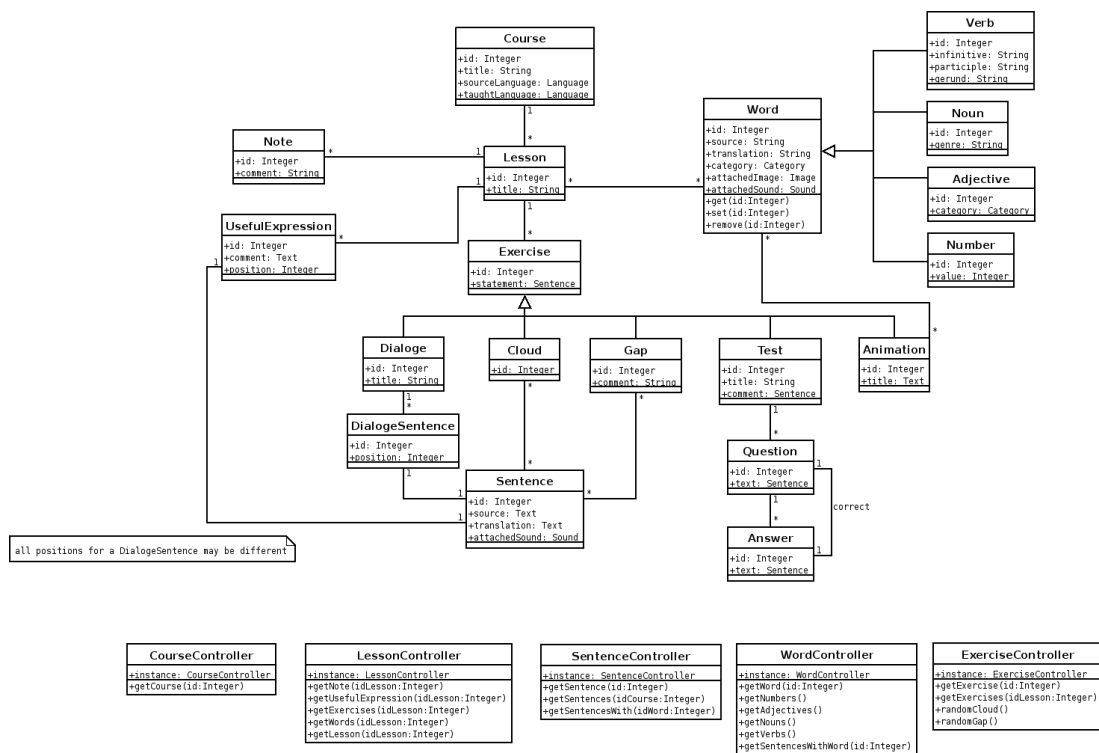


Figura A.1: Diagrama de clases de la primera y segunda iteración

# Apéndice B

## Vista final de la aplicación

A continuación se muestran una serie de imágenes correspondientes a las capturas de pantalla de la aplicación en su entorno de pruebas.

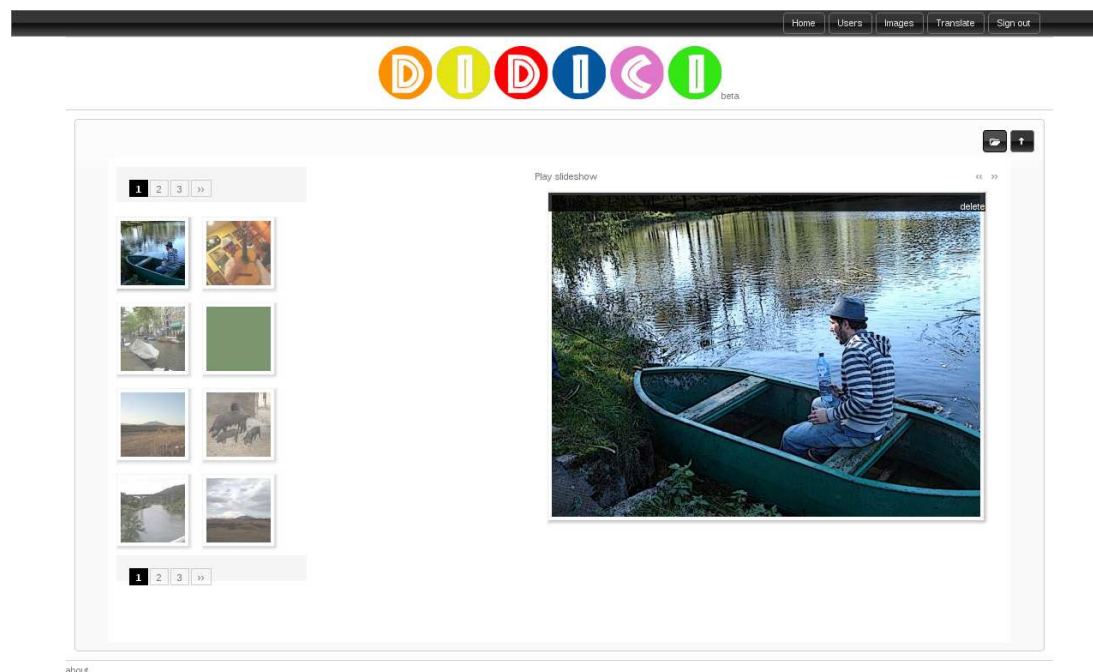
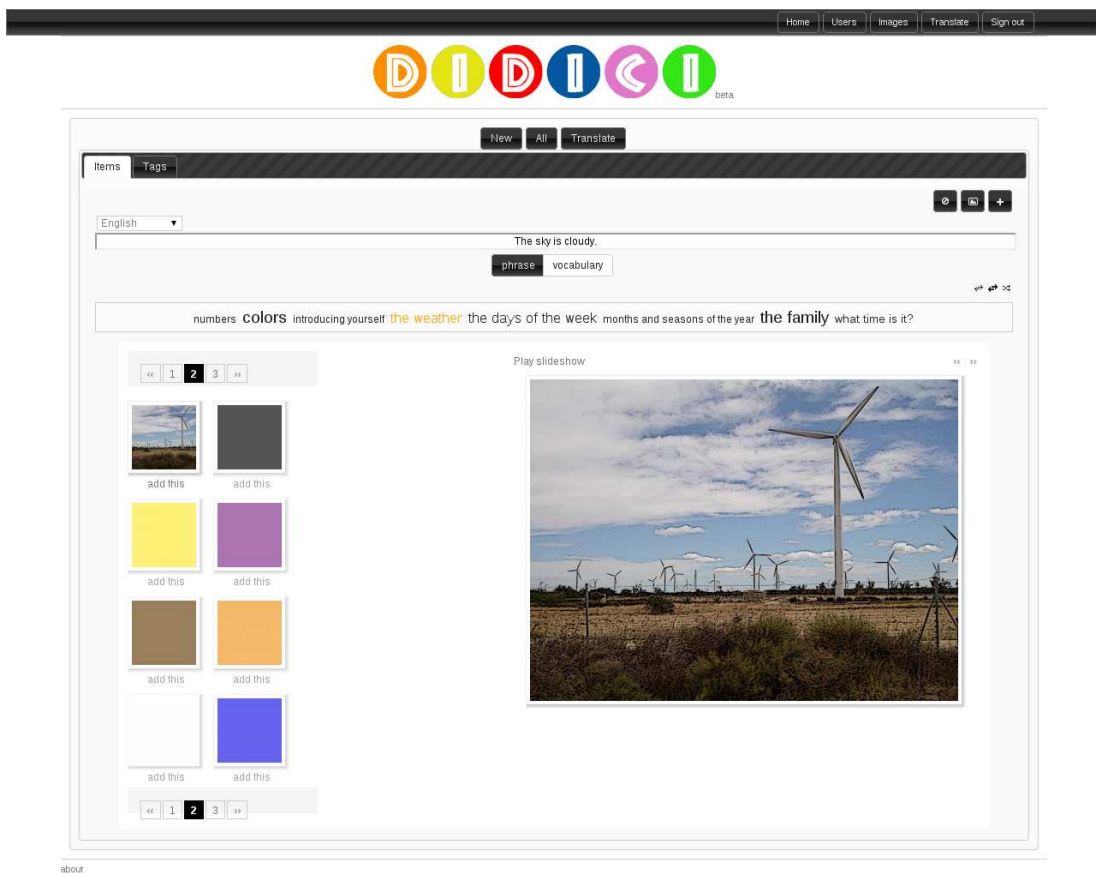


Figura B.1: Vista de la zona de imágenes

Figura B.2: Vista del formulario *nuevo ítem*.

Home Users Images Translate Sign out

D I D I C I beta

New All Translate

Català

Items Tags

Show 10 entries

English	Català	Audio	Record	Saved
April	abril		<input checked="" type="checkbox"/>	
August	agost		<input checked="" type="checkbox"/>	
black	negre		<input checked="" type="checkbox"/>	
blue	blau		<input checked="" type="checkbox"/>	
brown	marró		<input checked="" type="checkbox"/>	
December	desembre		<input checked="" type="checkbox"/>	
deep	profund		<input checked="" type="checkbox"/>	
eight	vuit		<input checked="" type="checkbox"/>	
eighteen	divuit		<input checked="" type="checkbox"/>	
eighty	vuitanta		<input checked="" type="checkbox"/>	

Showing 1 to 10 of 118 entries

about

Figura B.3: Vista de la zona de traducció

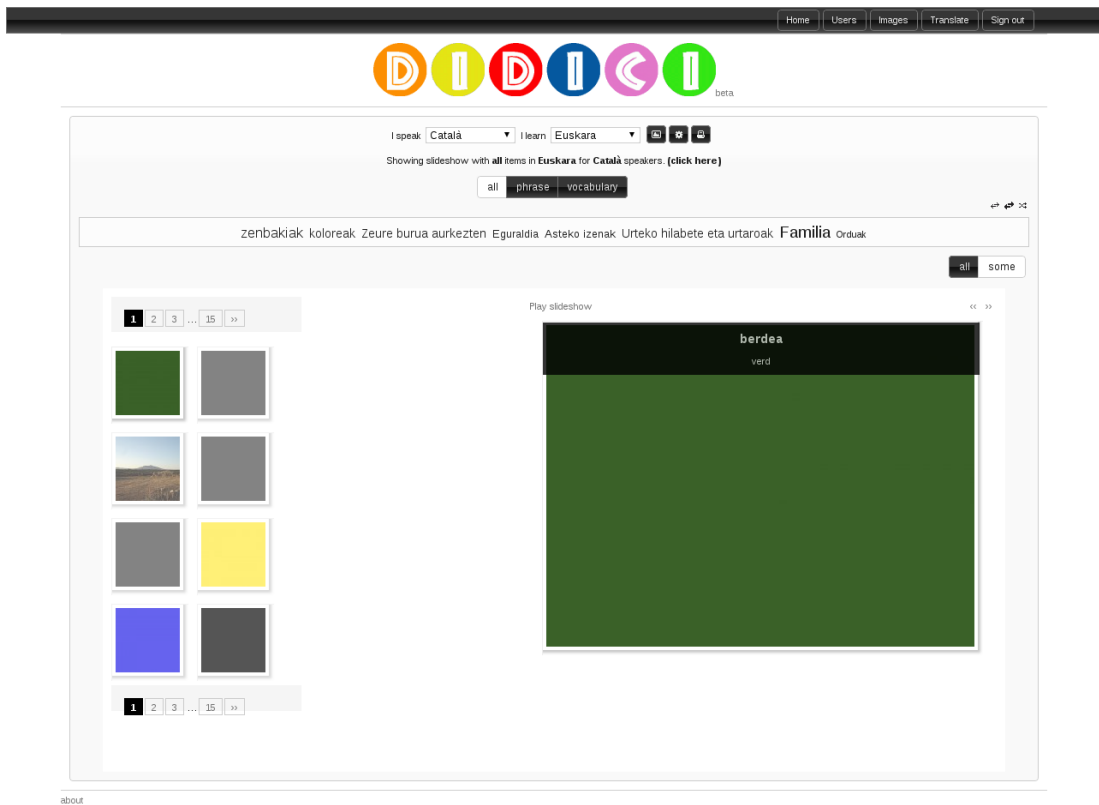


Figura B.4: Vista principal de la aplicación.



The screenshot displays a language learning application interface. At the top, there is a navigation bar with links for Home, Users, Images, Translate, and Sign out. Below this is a logo consisting of six colored circles (orange, yellow, red, blue, purple, green) containing the letters 'D', 'I', 'D', 'I', 'C', 'I' respectively, with the word 'beta' underneath. The main content area features a header with language selection: 'I speak: Deutsch' and 'I learn: Nederlands'. Below this, it states 'Showing exercises with all items in Nederlands for Deutsch speakers. (click here)'. There are filters for 'all', 'phrase', and 'vocabulary'. A search bar contains the text 'nummers kleuren jezelf voorstellen het weer de dagen van de week maanden en seizoenen de familie hoe laat is het?'. Below the search bar, there are navigation controls for 'Exercise 1 of 122' and a '?' icon. The exercise content includes three sentences with gaps to be filled:

- Gestern hatte ich Bauchweh.  
 had ik
- Es ist zwölf Uhr.  
 is
- Hallo! Wie geht es Ihnen?  
 Hoe

At the bottom of the exercise area, there is a text box containing the Dutch translation: 'Hallo! Gisteren Het het? buikpijn. gaat een twaalf uur.'

about

Figura B.5: Vista de los ejercicios interactivos. Ejercicio tipo *rellenar huecos* (*Filling the gaps*).

Home Users Images Translate Sign out

**D I D I C I** beta

I speak English I learn Português

Showing exercises with all items containing some of this tags **números, cores** in Portuguese for English speakers. [\(click here\)](#)

all phrase vocabulary

números cores introduzindo-se o tempo os dias da semana meses e estações do ano a família Que horas são?

all some

← Exercise 15 of 40 →

?

sixty

e e s a s t n s

about


Figura B.6: Vista de los ejercicios interactivos. Ejercicio tipo *Ordenar letras (Sort letters)*.

The screenshot displays a web-based language learning interface. At the top, there is a navigation bar with links for Home, Users, Images, Translate, and Sign out. Below this is a logo consisting of six colored circles (D, I, D, I, C, I) with the word 'beta' underneath. The main content area features a language selection menu where 'I speak Italian' and 'I learn Français' are chosen. A message indicates that exercises are filtered by the tag 'vous présenter in Français for Italian speakers'. There are filters for 'all', 'phrase', and 'vocabulary'. A list of tags includes 'numéros', 'couleurs', 'vous présenter', 'la météo', 'les jours de la semaine', 'mois et des saisons de l'année', 'la famille', and 'quelle heure est-il?'. The current exercise is 'Exercise 2 of 12', which involves sorting the sentence 'Ciao, mi chiamo Quizzy e ho un anno di età.' using the provided words: 'suis', 'Quizzy', 'd'un', 'je', 'm'appelle', 'je', 'âgé', 'et', 'Bonjour,', 'an.'. The interface also includes a small 'about' link at the bottom left.

Figura B.7: Vista de los ejercicios interactivos. Ejercicio tipo *Ordenar frase (Sort sentence)*.

The screenshot displays the application's interface for an interactive exercise. At the top, there is a navigation bar with links for Home, Users, Images, Translate, and Sign out. Below this is a logo consisting of six colored circles (D, I, D, I, C, I) with the word 'beta' underneath. The main content area features a header with language selection options: 'I speak: العرب' and 'I learn: Català'. A message states: 'Showing exercises with all items in Català for العرب speakers. [click here]'. Below this are filters for 'all', 'phrase', and 'vocabulary'. A horizontal menu lists various topics: 'números', 'colors', 'presentant-te', 'el temps', 'els dies de la setmana', 'mesos i estacions de l'any', 'la família', and 'quina hora és?'. The exercise is titled 'Exercise 1 of 12'. The question is 'مرحبا! كيف حالك?' (Hello! How are you?). Below the question are three empty text input fields. The second field is labeled 'اليوم منضم' (Today is) and the third is labeled 'مرحبا! انا بفصل غرامة وأنت?' (Hello! I am in a fine and you?). At the bottom, there is a list of words to be used: 'gràcies,', 'Hola!', 'Estic', 'i', 'Avui', 'sol.', 'bé,', 'fa', 'Com', 'tu?', 'Hola!', 'estàs?'. A small 'about' link is visible at the bottom left.

Figura B.8: Vista de los ejercicios interactivos. Ejercicio tipo *Ordenar frases* (*Sort sentences*).



Exercise 1

fytfi

Exercise 2

the afirteh

Exercise 3

eufentor

Exercise 4

oen nuthsoda

Exercise 5

\_\_\_\_\_ is \_\_\_\_\_

\_\_\_\_\_ \_\_\_\_\_ Barcelona.

\_\_\_\_\_ was cloudy all \_\_\_\_\_

\_\_\_\_\_ I sky am long. sunny. The Today day from

Exercise 6

\_\_\_\_\_ \_\_\_\_\_ Barcelona.

\_\_\_\_\_ is \_\_\_\_\_

Figura B.9: Vista de los *ejercicios en PDF*.



# Bibliografía

- [1] National Geographic Society's Enduring Voices Project  
<http://travel.nationalgeographic.com/travel/enduring-voices/>
- [2] Método volere <http://www.volere.co.uk/template.htm>
- [3] Open Source <http://www.opensource.org/docs/osd>
- [4] ISO639-2 <http://www.loc.gov/standards/iso639-2/>
- [5] Ley Orgánica de Protección de Datos <http://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf>
- [6] Dia Diagram Editor <http://projects.gnome.org/dia/>
- [7] Balsamiq Mockups <http://builds.balsamiq.com/b/mockups-web-demo/>
- [8] tFPDF <http://fpdf.org/en/script/script92.php>
- [9] Galleriffic <http://www.twospys.com/galleriffic/>
- [10] Flowplayer <http://flowplayer.org/>
- [11] upload.class.php <http://www.verot.net/php-class-upload.htm>
- [12] datatables <http://datatables.net>





# Glosario

Babbel	es una plataforma de pago dedicada a la enseñanza de idiomas online desde Enero de 2008., 1
bugs	es el resultado de un fallo o deficiencia durante el proceso de creación de programas de software., 3
Dia Diagram Editor	es una aplicación informática de propósito general para la creación de diagramas, 9
Drupal	es un sistema de gestión de contenido modular multipropósito y muy configurable que permite publicar contenidos, almacenarlos en una base de datos y editarlos utilizando un entorno Web, 1
Internet	es un conjunto descentralizado de redes de comunicación interconectadas a nivel mundial, 1, 6
Joomla	es un sistema de gestión de contenidos, y entre sus principales virtudes está la de permitir editar el contenido de un sitio web de manera sencilla, 1
Livemocha	es una plataforma comercial online de aprendizaje de idiomas en comunidad, cuenta con material en 38 idiomas y una plataforma para la interacción entre usuarios donde se ayudan entre ellos a aprender idiomas., 1
Ley Orgánica de Protección de Datos (LOPD)	es una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar, 8

Moodle	es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea, 1
MySQL	es un sistema de gestión de base de datos relacional, multihilo y multiusuario, 27
navegador web	es un programa que permite ver la información que contiene una página web (ya se encuentre ésta alojada en un servidor dentro de la World Wide Web o en un servidor local), 6, 7
Open Source	es el término con el que se conoce al software distribuido y desarrollado libremente, 6, 27
PDF	Portable Document Format, es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems, 7
PHP Hypertext Pre-processor (PHP)	es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas, 27
páginas web	es un documento electrónico adaptado para la World Wide Web que generalmente forma parte de un sitio web. Su principal característica son los hipervínculos de una página, 1, 27
Roseta Stone	es un programa para aprender idiomas que utiliza una combinación de imágenes, texto y sonido, donde los niveles de dificultad suben a medida que el estudiante avanza. En estas lecciones el estudiante aprende vocabulario y funciones gramaticales sin traducción alguna., 1
Unified Modeling Language (UML)	es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema, 9