

Memoria proyecto:

TradúceMe

**C.I.F.P. “JUAN DE COLONIA”**

**Departamento de Informática**

**2º DAW**

**Ángel Mori Martínez Díez**

**Jueves, 3 de junio de 2021**

# Contenido

[1. Introducción 6](#_Toc73034744)

[1.1. Descripción 6](#_Toc73034745)

[1.2. Justificación 8](#_Toc73034746)

[2. Planificación 9](#_Toc73034747)

[2.1. Requisitos 9](#_Toc73034748)

[2.1.1. Requisitos funcionales 9](#_Toc73034749)

[2.1.2. Requisitos no funcionales 10](#_Toc73034750)

[2.2. Recursos 10](#_Toc73034751)

[2.2.1. Recursos hardware 10](#_Toc73034752)

[2.2.2. Recursos software 11](#_Toc73034753)

[2.3. Planificación temporal 11](#_Toc73034754)

[2.4. Planificación económica 14](#_Toc73034755)

[3. Tecnologías 15](#_Toc73034756)

[3.1. HTML 15](#_Toc73034757)

[3.2. CSS 15](#_Toc73034758)

[3.3. JavaScript, AJAX, jQuery 16](#_Toc73034759)

[3.4. JSON 17](#_Toc73034760)

[3.5. Bootstrap 17](#_Toc73034761)

[3.6. Apache HTTP Server 18](#_Toc73034762)

[3.7. MySQL 18](#_Toc73034763)

[3.8. PHP 18](#_Toc73034764)

[3.9. TinyMCE 19](#_Toc73034765)

[4. Desarrollo y secuenciación temporal 20](#_Toc73034766)

[4.1. Diseño 20](#_Toc73034767)

[4.1.1. Diagrama de aplicación 20](#_Toc73034768)

[4.1.2. Diagrama E/R (entidad/relación) y esquema de BBDD 21](#_Toc73034769)

[4.1.3. Diseño de interfaces 22](#_Toc73034770)

[4.2. Pruebas 29](#_Toc73034771)

[*4.2.1.* Pruebas de estilo 29](#_Toc73034772)

[4.2.2. Pruebas de funcionalidad 29](#_Toc73034773)

[5. Conclusiones finales 31](#_Toc73034774)

[5.1. Grado de cumplimiento de los requisitos fijados 31](#_Toc73034775)

[5.2. Propuestas de mejora o ampliaciones futuras 31](#_Toc73034776)

[6. Guías 32](#_Toc73034777)

[6.1. Guía de uso 32](#_Toc73034778)

[6.2. Guía de instalación 36](#_Toc73034779)

[7. Referencias 37](#_Toc73034780)

# Índice de figuras

Fig. 1‑1-Pantalla inicial de la aplicación 5

Fig. 1‑2-Pantalla de la sección de información del modo administrador 6

Fig. 4‑1-Diagrama de la interfaz de usuario 19

Fig. 4‑2-Diagrama del modo administrador 19

Fig. 4‑3- Diagrama E/R de la base de datos 20

Fig. 4‑4- Pantalla inicial de la aplicación en la versión con el menú desplegado 21

Fig. 4‑5- Pantalla inicial de la aplicación en la versión con el menú sin desplegar 22

Fig. 4‑6- Cabecera en inglés 23

Fig. 4‑7-Cabecera con desplegable 23

Fig. 4‑8- Banderas para al cambio de idioma 23

Fig. 4‑9- Lista de servicios ofrecidos y carrusel de opiniones 24

Fig. 4‑10- Sobre mí 24

Fig. 4‑11-Servicios de traducción 25

Fig. 4‑12- Contacto 25

Fig. 4‑13- Modelo de página 26

Fig. 4‑14- Accesos al modo administrador 26

Fig. 4‑15- Añadido de páginas: editando página 27

Fig. 6‑1- Añadido de páginas 31

Fig. 6-2- Mensaje de éxito 32

Fig. 6-3- Página de edición: eligiendo 32

Fig. 6-4- Mensaje de error al editar una pagina 32

Fig. 6‑5- Página de edición y comparación 32

Fig. 6‑6- Mensajes la editar dos páginas 33

Fig. 6‑7- Parte superior de página de información 34

Fig. 6‑8- Parte inferior de página de información 35

# Índice de tablas

[Tabla 2‑1: Planificación temporal del proyecto 7](#_Toc477346555)

# Introducción

## Descripción

El proyecto TradúceMe consiste en una aplicación web de adaptable a todo tipo de dispositivos, personalizable, ampliable y administrable. Abreviando: una página de aterrizaje con un gestor de contenidos (CMS). Esta aplicación ha sido desarrollada con JavaScript, PHP, una base de datos MySQL, HTML y CSS, ayudadas estas dos últimas tecnologías por la biblioteca Bootstrap de código abierto y multiplataforma para el diseño de sitios y aplicaciones web. También, se ha usado puntualmente JSON en la memoria caché para la transferencia de datos desde PHP y MySQL a JavaScript.

La interfaz de usuario es la esperada de una página de aterrizaje, motivo por el cuál la estructura y las páginas principales le resultarán familiares al usuario habituado a internet: inicio, sobre mí, servicios ofrecidos y contacto. Además, las páginas pueden estar disponibles tanto en inglés como en español.

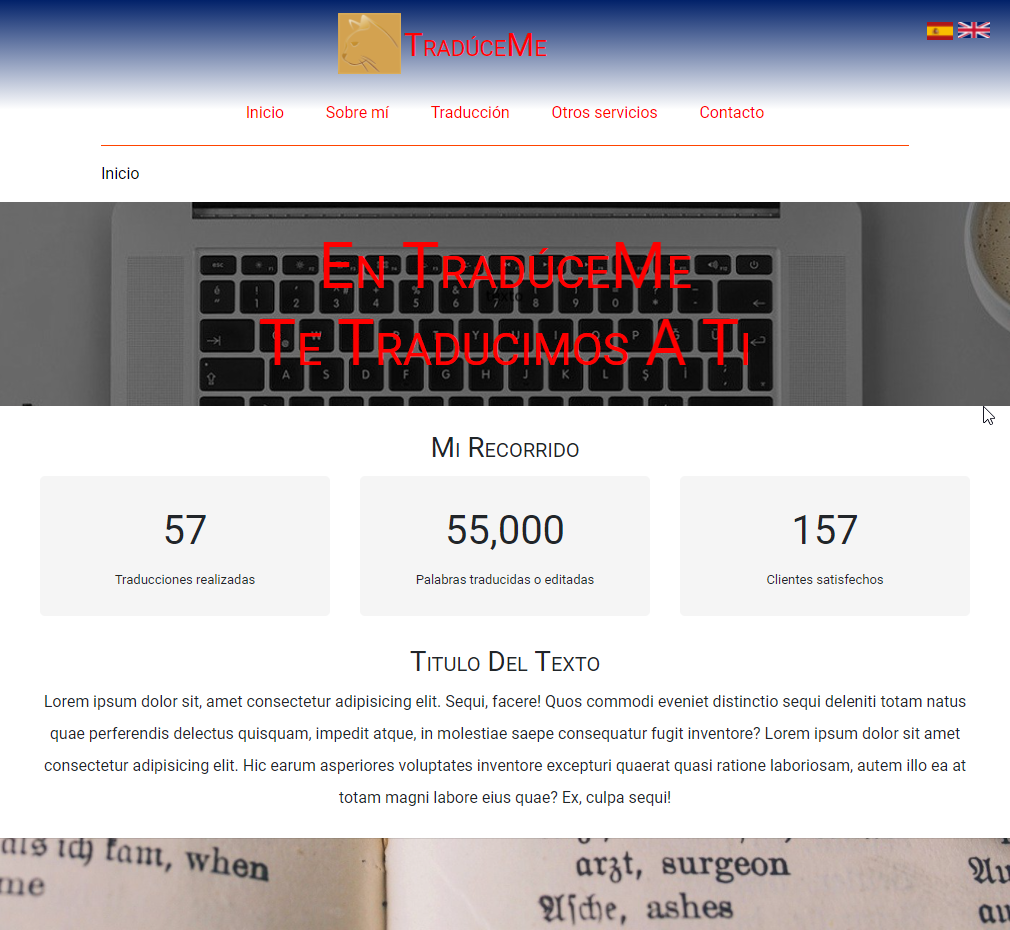


Fig. ‑-Pantalla inicial de la aplicación

Como decía al principio, la página es personalizable, ampliable y administrable. ¿Exactamente qué quiere decir esto? Existe un modo administrador en el que es posible:

* Añadir nuevas páginas a la aplicación, accesibles desde su propio enlace.
* Editar o comparar y editar las páginas en los distintos idiomas.
* Visualizar la cantidad y el nombre de páginas creadas, para, por ejemplo, saber qué páginas no están todavía traducidas.
* Borrar la memoria caché.
* Ver los accesos y los intentos de acceso modo administrador.

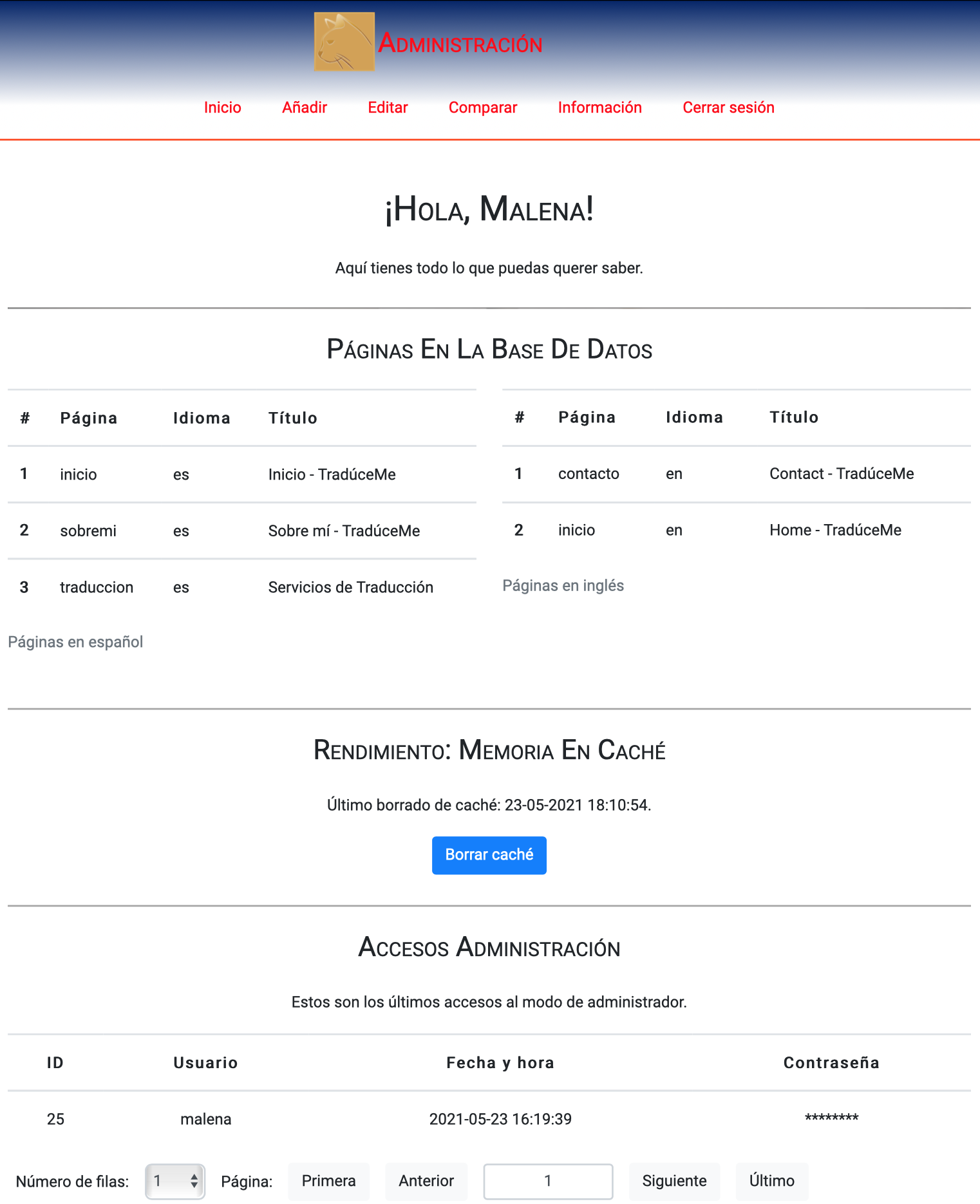


Fig. ‑-Pantalla de la sección de información del modo administrador

Cabe destacar que el objetivo del proyecto no es crear la página de aterrizaje al completo, sino preparar el marco donde la traductora pueda hacerlo, personalizando el contenido cuando y como considere a través del gestor de contenidos disponible desde el modo administrador.

## Justificación

TradúceMe busca dar a una traductora, llamada Malena, una página de aterrizaje donde permitirla explicar a qué se dedica, qué servicios ofrece y quién es ella. Así como crear nuevas páginas según necesite. Permitiendo de esta manera, tener un espacio propio en internet en el que darse visibilidad.

La profesión de traductor es una de las profesiones más antiguas del mundo trabajo primordial para la difusión lingüística, estética e ideológica a lo largo de los siglos. Malena es una emprendedora que inicia su vida laboral como traductora.

Hoy en día, todo profesional que se precie ha de disponer de una página web que le describa y que proporcione las herramientas precisas para contratar sus servicios

Este espacio en internet, en primer lugar, permite que te encuentren fácilmente y sin conocerte de antemano, y, en segundo, se traduce —nunca mejor dicho— en que el cliente tendrá un mayor sentimiento de confianza a la hora de decidir a quién contratar los servicios. Una página web es uno de los más importantes activos que todo negocio tiene que tener.

Era necesario también tanto que la aplicación estuviese en inglés y español —dada la naturaleza de su trabajo—, como que fuese ampliable y, sobre todo, editable. Estas dos últimas funciones permiten actualizar el contenido fácilmente con tan sólo iniciando sesión en el modo administrador, facilitando traducir las páginas y crear contenido nuevo.

# Planificación

## Requisitos

### Requisitos funcionales

De la interfaz de usuario:

* Permitir acceder al contenido en español y en inglés cuando este esté disponible en ambos idiomas.
* Mostrar el contenido directamente en el idioma disponible cuando no lo esté en el idioma que se está usando.
* Detectar las preferencias de idioma del navegador del usuario, mostrándole el idioma con mayor preferencia configurada (solamente entre inglés y español).

Del modo administrador:

* Permitir añadir páginas a la base de datos y hacerlas funcionales y accesibles desde la aplicación.
* Permitir editar las páginas, el píe de página y la cabecera tanto en HTML como con una interfaz visual de edición de HTML con los estilos aplicados.
* Permitir la edición tanto individual como simultánea de las versiones española e inglesa de las páginas, el pie de página y la cabecera.
* Cuando una página exista solamente en un idioma, facilitar la creación y traducción de la página al idioma no existente, ofreciendo la posibilidad de editarla (aunque no exista en la base de datos). Esto se hará duplicando el contenido de la página del idioma existente, de forma que solo habría que traducir el contenido.
* Permitir borrar la caché, tanto cuando se guarda una modificación en la base de datos, tanto cuando se considere.
* Mostrar la lista páginas existentes en la base de datos en cada uno de los idiomas.
* Mostrar los accesos al modo de administrador y actualizando el contenido cada vez que se navega por la lista de accesos sin tener que recargar la página. Esto incluye que se muestren los nuevos accesos que se puedan producir mientras se navega por la lista.
* Facilitar al máximo la ampliación, personalización y edición de la aplicación web por parte de la traductora de forma independiente.

De uso de datos, eficiencia y seguridad:

* Guardar la siguiente información correctamente y de forma recuperable en la base de datos: las páginas, los pies de página, las cabeceras, los datos de acceso de los administradores y los intentos de acceso al modo administración.
* Al acceder a una página de la aplicación mediante la interfaz de usuario, comprobar la existencia en caché de la página a la que se quiere acceder y las versiones en ese idioma del pie de página en la cabecera.
  + Si no existen, acceder a la base de datos y guardarlos en caché. Después, mostrar la página normalmente.
  + Si existen, mostrar la página.
* Controlar el acceso a al modo administrador mediante un inicio de sesión con usuario contraseña y permitir cerrar la sesión de administrador sin tener que cerrar el navegador.
* No almacenar jamás las contraseñas en texto plano. Salvo las de los accesos fallidos al modo administrador, de forma que desde este se puedan ver los usuarios y contraseñas con los que se ha intentado acceder, facilitando así la detección de amenazas.
* Evitar el acceso no deseado a las carpetas de todo el árbol de directorios de la aplicación.
* Reducir al mínimo la cantidad de archivos de SS y JavaScript.

### Requisitos no funcionales

* Ser intuitiva tanto para los usuarios como para los administradores, y no desviarse demasiado de los estándares a los que están acostumbrados.
* Disponer de un diseño que muestre la información de las distintas páginas de la aplicación de forma ordenada y comprensible.
* Ser adaptable a los distintos tipos de dispositivos de pantalla rectangular que hay actualmente en el mercado o pueda haber en el futuro.

## Recursos

### Recursos hardware

* Ordenador de sobremesa o portátil.
* Distintos dispositivos para visualizar la interfaz de la aplicación:
  + Ordenadores
  + Móviles
  + Tabletas
  + *E-Books*

### Recursos software

Lista de recursos software

* XAMPP:

“Es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl” [[1]](#endnote-1).

* Visual Studio Code

Editor de código fuente desarrollado por Microsoft [[2]](#endnote-2).

* Git, GitHub y GitHub Desktop:

Git es un Sistema de control de versiones libre y de código abierto [[3]](#endnote-3); GitHub una plataforma para alojar proyectos con el uso de Git; y GitHub Desktop es la aplicación de escritorio del anterior [[4]](#endnote-4).

* Navegadores:
  + Chrome
  + Opera
  + Safari
  + Mozilla Firefox
  + Microsoft Edge

Necesarios para comprobar el buen diseño de la aplicación

* Inkscape:

“Es un editor de gráficos vectoriales libre y de código abierto”[[5]](#endnote-5).

## Planificación temporal

La fase inicial no es otra que la del **prototipado de la aplicación web**. Para esto es necesario hablar con la traductora para ver qué es lo que busca obtener de la aplicación web, aconsejarla según mis conocimientos de las tecnologías y del sector y, finalmente, realizar algunos bocetos básicos de las distintas partes y páginas de la web. Dibujar y redibujar hasta obtener su visto bueno.

La segunda fase del proyecto es la del **diseño y relleno de la interfaz de usuario**. La cual consiste en varias tareas de diseño —HTML + CSS— que se llevarán a cambio simultáneamente y de forma complementaria. Diseño de estilos, colores, fuentes y estructura. Esto es posible subdividirlo según las diferentes partes de la web:

* Cabecera y el pie de página en versión móvil, adaptable a versión escritorio.
  + Esto es especialmente importante ya que la cabecera contendrá el menú de navegación de la aplicación web.
* Contenido de las páginas:
  + Migas de pan
  + Contendido diferenciado de cada una de las páginas de la aplicación.
* Botones de cambio de idioma.

La tercera fase es en la que entran en el juego PHP y las bases de datos: **generación dinámica de la web**. Ya tendríamos en este punto todo el HTML y CSS y habríamos confirmado con la traductora que lo que tenemos es lo que quiere, y en caso contrario, habríamos hecho las modificaciones pertinentes.

Para generar la web con PHP debemos usar un servidor capaz de ejecutar los archivos escritos en este lenguaje y conectar con bases de datos: XAMPP. Se debe diseñar la base de datos de forma que la interfaz de usuario pueda seguir una estructura MVC —modelo, vista, controlador—. Así mismo, seguir la siguiente estructura para el modelo MVC:

* Modelo:
  + Es la capa que trabaja con los datos. Accede a la información de la base de datos para editarla, tratarla, mostrarla o asegurarla.
  + Habrá varias clases para esta finalidad.
    - Solamente una interactuará con la base de datos.
    - El resto solo tratará los datos que esa clase les proporcione y los datos de los formularios.
  + Existirán varias clases de excepciones.
* Vista:
  + Una única plantilla para todas las páginas de la interfaz de usurario.
  + Recibe los datos a mostrar y genera las distintas páginas de la aplicación web.
* Controlador:
  + Un controlador principal encargado de acceder a los datos con la ayuda de los modelos. Debe:
    - Controlar la cookie de idioma y el cambio de idioma.
    - Pasar a la vista los datos necesarios para mostrar las páginas correctamente.
  + Un controlador por cada página accesible a través de la interfaz de usurario. Comparte información de la página a mostrar con el controlador principal.

Fase cuatro: **desarrollo modo administrador**. Debe permitir añadir páginas, editar las cabeceras, los pies de página y el contenido de las páginas y mostrar información de la web. Para esto hay que crear las secciones de:

* Añadido de páginas.
* Edición de páginas
* Edición y comparación de páginas en los dos idiomas.
* Visualización de las páginas guardadas en la base de datos.

Además, en esta fase es necesario crear una página de inicio de sesión con usuario y contraseña y ampliar la base de datos para que almacene la información de los usuarios que pueden acceder a este modo administrador.

**Caché y seguridad** es la quinta fase. Es Las páginas se generaban dinámicamente a partir de los datos de la base de datos cada vez que un usuario accede a una página. No hay eficiencia en esto. A pequeña escala no es un inconveniente, pero con muchos usuarios accediendo a la aplicación, se sobrecargaría el servidor que alojase la web —en este caso XAMPP—, ralentizando la web.

Esta fase del proyecto es perfectamente omisible y podría ir en el apartado de *propuestas de mejora o ampliación* en caso de que se alarguen demasiado las fases anteriores, dejando sin tiempo para realizar esta. Sin embargo, en un principio forma parte de la planificación inicial.

La aplicación web con las cuatro primeras fases ya sería perfectamente funcional. La quinta fase modifica cosas terminadas de las fases anteriores. Estos son los cambios a introducir:

* La vista solo utilizará nombre la página que tiene que mostrar y el idioma en que hacerlo —obteniendo esto del controlador— ya que en la caché encontrará las variables y los archivos necesarios.
* El controlador comprueba se asegura de que el árbol de la caché esté correctamente creado.
* El controlador, antes de acceder a la base de datos a por la cabecera, el pie de página y el contenido de la página, comprueba si estos datos están ya en la caché en forma de archivo.
  + Si no lo están, consigue los datos y los guarda como archivos en la caché.
  + Si ya estaban simplemente omite este paso.

Última fase: **detalles y arreglos**. Si es posible y sobra tiempo, en esta fase, se retocarán cositas, detalles. Como, por ejemplo, el uso del archivo *.htaccess* o la creación de una página para clásico error 404.

Como la fase cinco es bastante más relevante, tendrá prioridad sobre esta. Por este motivo, se irá viendo según de desarrolle el proyecto qué cosas sería conveniente o no priorizar o hacer en esta última fase. Si es que llegase el momento de siquiera planteárselo.

En la siguiente tabla (Tabla 2.1) se detalla la planificación temporal inicial del proyecto:

Tabla ‑: Planificación temporal del proyecto

|  |  |
| --- | --- |
| **Descripción de la tarea** | **Nº de horas** |
| Prototipado de la aplicación web | 3 |
| Diseño y relleno de la interfaz de usuario | 15 |
| Generación dinámica de la web | 15 |
| Desarrollo modo administrador | 12 |
| Caché y seguridad | 7 |
| Detalles y arreglos | ≥ 3 |
| **TOTAL HORAS** | ≥ 55 |

## Planificación económica

En un principio el no hay ningún presupuesto para realizar el proyecto.

Todo el software que se va elegido usar es gratuito. Motivo por el cuál es este el software que se va a emplear y no otro. Parte de este conjunto de software es, además, software libre, lo cual es un plus ya que está respaldado por toda una comunidad.

En cuanto a hardware, no se requiere comprar nada, por lo que tampoco

# Tecnologías

Las tecnologías utilizadas son bastante habituales en el desarrollo web. Algunas de ellas incluso esenciales. Son las siguientes: HTML, CSS, Bootstrap, JavaScript, AJAX, jQuery, JSON, servidor HTPP Apache, PHP y MySQL.

## HTML

Sus siglas significan lenguaje de marcado de hipertexto. Se ha usado el estándar HTML5. Versión, que al igual que las anteriores, está a cargo del Consorcio WWW, también conocido como W3C.

A su vez, la versión HTML5 posee una mayor compatibilidad con la mayoría de navegadores respecto a su versión anterior. Además, es más sencilla y tiene nuevas etiquetas. Como consecuencia de todo esto encontramos una carga más rápida de las páginas web.

Es el idioma en el que se escriben las páginas de Internet. No existiría la gran si HTML no existiese. Es un lenguaje reconocido universalmente que permite compartir información de forma global.

Está basado en etiquetas contenedores de otras etiquetas o texto simple. Éstas etiquetas pueden tener a su vez atributos con o sin valores, deben comenzar y acabar con los caracteres de menor y mayor que, “<” y “>”, respectivamente.

En su gran mayoría, las etiquetas después de abriese deben después cerrarse. Esto se hace escribiendo la etiqueta con el nombre precedido por el carácter de la barra inclinada, “/”. Unas pocas en cambio no necesitan ser cerradas.

Las etiquetas de este lenguaje están pre definidas por el estándar. Cada una tiene una finalidad propia y única.

## CSS

Son las conocidas como hojas de estilo en cascada por el significado de sus iniciales en inglés. También es el W3C quien se encarga de desarrollar este estándar.

Su finalidad es estructurar las páginas dando estilos a los documentos desde fuera de las etiquetas HTML consiguiendo con esto una mayor limpieza en este código, haciéndolo más comprensible. El CSS podría estar incluido en la etiqueta de HTML “*style*” o podría estar incluso fuera de los archivos que lo contengan.

CSS presenta multitud de características bastante útiles:

* Permite tanto estructurar toda una aplicación web desde un mismo archivo CSS cómo aplicar varias hojas de estilos CSS a un mismo documento HTML
* Las hojas de estilo pueden ser aplicadas por los navegadores, los desarrolladores de las aplicaciones web o por los usuarios.
* Es posible diferenciar los estilos a aplicar dependiendo de muchos factores. Por ejemplo:
  + - Si se va a imprimir o a mostrar por pantalla.
    - Si el dispositivo está en horizontal o en vertical
    - Si tiene un ancho o un alto en algún rango determinado.

## JavaScript, AJAX[[6]](#endnote-6), jQuery[[7]](#endnote-7)

JavaScript es un lenguaje interpretado, basado en prototipos, multiparadigma, de un solo hilo con soporte de programación orientada a objetos. Es un dialecto del estándar ECMAScritp, ejecutado generalmente en el lado del cliente, es decir, en el navegador del usuario.

Interactúan con las páginas webs gracias a la implementación del DOM —Modelo de Objetos del Documento.

JavaScript permite ejecutar acciones cuando el usuario realiza determinadas acciones, conocidas como eventos.

Cuando un evento tiene lugar, y si así está programado, se ejecutan funciones concretas. Lo cual da lugar a infinidad de posibilidades de personalización, haciendo dela navegación web una experiencia mucho más interactiva que simplemente ver el contenido de las páginas de la aplicación web.

jQuery por su parte, es un complemento a JavaScript muy extendido en el mundo del desarrollo web, ya que ayuda a reducir código. Tiene también sistemas de plantillas y *plugins*.

AJAX, que con su acrónimo traducido al español significa JavaScript asíncrono y XML, es una técnica que permite leer datos del servidor una vez cargada la página, actualizar esta sin recargarla y enviar datos al servidor web en segundo plano.

Esto permite que se creen aplicaciones más rápidas y con una mejor respuesta al usuario y las acciones que realiza.

No es una tecnología por sí misma, es un modo de utilizar algunas tecnologías existentes, incluyendo: HTML, CSS, JavaScript, DOM, XML y el objeto XMLHttpRequest, que es lo más importante, ya que es el encargado de intercambiar datos con el servidor web en segundo plano.

## JSON

El nombre de JSON viene de Notación de Objeto de JavaScript. En un origen era una transliteración de los objetos de JavaScript, sin embargo, es considerada ya un formato independiente de este lenguaje.

La idea principal de JSON es ser un formato de intercambio de datos. Realiza su función incluso entre distintos lenguajes de programación. Ya que está tan extendido que la gran mayoría son capaces de pasar sus datos al formato de este lenguaje.

Este hecho y no otro ha sido el motivo por el que se ha usado JSON en este proyecto.

Cabe destacar que el contenido de los archivos JSON es muy fácil de leer por los humanos y que el tamaño de los archivos es mucho menor que el de otros lenguajes de transferencia de datos como puede ser XML.

## Bootstrap[[8]](#endnote-8)

Bootstrap es un *framework* bastante simple —en lo que respecta a facilidad de uso— que facilita todo el proceso de diseño web.

Crea interfaces de usuario que se adaptan a las diferentes plataformas, navegadores y dispositivos. Ayuda a ahorrar mucho tiempo en el diseño y la maquetación de las interfaces.

Desarrollado por Twitter y combinando CSS y JavaScript, cuando el código de Bootstrap fue liberado por los creadores, se convirtió en el proyecto colaborativo más grande de GitHub en 2012. Desde entonces se ha ido desarrollando y puliéndose haciéndose más útil, más sencillo y más eficiente.

Es posible usarlo mediante instalación o mediante referencia a sus librerías con un enlace. Esta última opción es la que hemos utilizado.

Emplea librerías JavaScript, como JQuery y funciona con todos los navegadores, incluido Internet Explorer, lo cual no deja de ser sorprendente. Casi tanto como el hecho de que este navegador se siga utilizando.

Dispone de distintos diseños predefinidos con estructuras que pueden disponer puntos de ruptura.

Bootstrap se implementa en la práctica haciendo uso de las clases de las etiquetas HTML. Da estilos determinados a clases determinadas.

## Apache HTTP Server[[9]](#endnote-9)

Es un software de servidor web multiplataforma gratuito y de código abierto. Desarrollado y mantenido por una comunidad abierta de desarrolladores

Apache HTTP Server fue lanzado en 1995 y lleva siento el servidor web más popular en internet desde abril de 1996.

Este tipo de servidores tienen un pequeño problema con las conexiones simultáneas cuando estas comienzan a sobrepasar la cifra de las 10.000. Sin embargo, la aplicación web realizada en este proyecto no alcanzará esta cifra, por lo que no habrá ningún problema.

## MySQL[[10]](#endnote-10)

MySQL es un sistema de gestión de bases de datos relacionales de código abierto, multihilo y multiusuario. Organiza los datos en una o más tablas que pueden o no estar relacionadas con otras tablas.

Es un software gratuito bajo la licencia GNU *General Public License* y la popularidad como aplicación web que tiene está bastante unida a lenguaje de programación web PHP, del que hablamos justo bajo este punto.

Este sistema basa su gestión en sentencias SQL, siglas del inglés que traducidas significan Lenguaje de Consulta Estructurada.

## PHP[[11]](#endnote-11)

PHP es un lenguaje interpretado de propósito general muy extendido que, concebido para desarrollo web, se puede incrustar en el código HTML.

La curva de aprendizaje de este lenguaje, aunque corta, permite crear aplicaciones de contenido dinámico complejas. Es ejecutado en el servidor web, de forma invisible para el navegador web y el usuario.

Está orientado a crear fácilmente interfaces de usuario. Hay multitud de *frameworks* de PHP que facilitan esto aún más, como puede ser Laravel, sin embargo, solo emplearemos PHP 7.

Se ha elegido la versión PHP 7 y no 8 debido a las incompatibilidades que presenta esta última versión el lenguaje al no ser tan reciente.

No hay que olvidarse de mencionar que PHP es de código abierto, motivo por el que es el lenguaje más popular de programación web.

## TinyMCE [[12]](#endnote-12) [[13]](#endnote-13)

TinyMCE es un editor visual libre y de código abierto desarrollado con enteramente con JavaScript.

Permite editar y crear código HTML de forma visual, como se haría en un editor de texto estilo *Libre Office*. Con la posibilidad de aplicarle estilos por medio de CSS.

# Desarrollo y secuenciación temporal

## Diseño

### Diagrama de aplicación

Distinguimos dos diagramas de aplicación: uno para el interfaz de usuario (Fig. 4-1) y otro del modo administrador (Fig. 4-2).

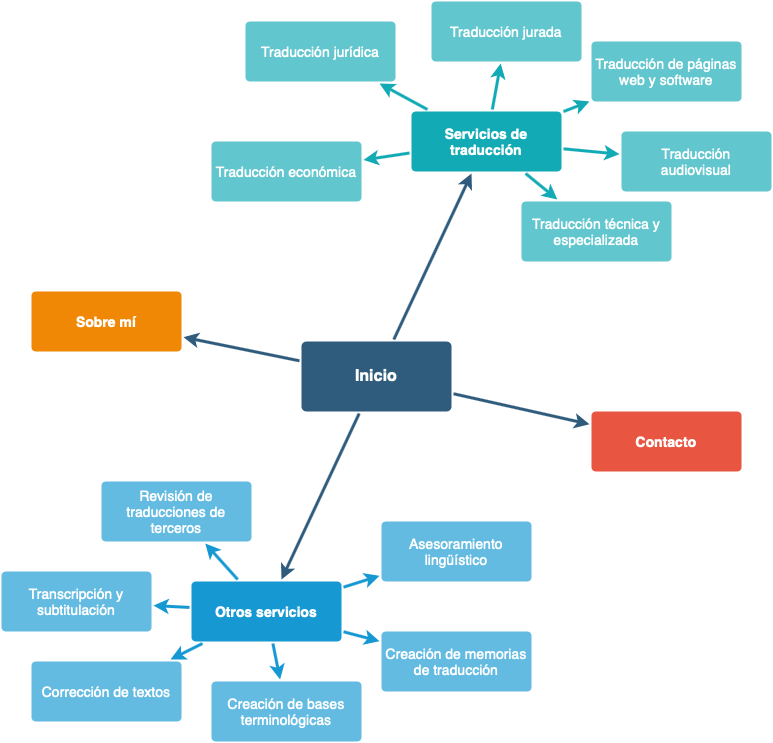


Fig. ‑-Diagrama de la interfaz de usuario

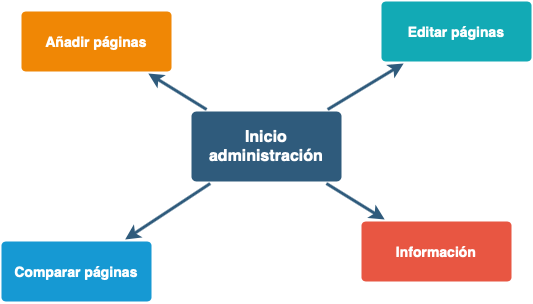


Fig. ‑-Diagrama del modo administrador

### Diagrama E/R (entidad/relación) y esquema de BBDD

En la siguiente figura (Fig. 4-3) se puede ver el diagrama entidad relación de la base de datos:

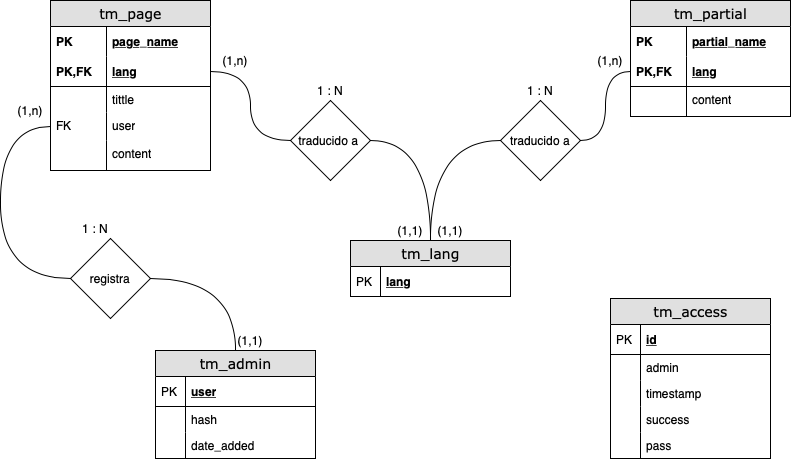


Fig. ‑- Diagrama E/R de la base de datos

Todas las tablas llevan un prefijo por seguridad. De esta forma es más complicado que alguien acceda a ellas al no saber el nombre.

La tabla *tm\_admin* contiene en sus registros los usuarios que pueden acceder al modo administrador, un *hash* de sus contraseñas y las fechas en las que fueron añadidos.

Por su parte, la tabla *tm\_access*, lleva un registro de los intentos de acceso al modo administrador. Tanto los exitosos como los fallidos. De estos últimos, guarda la contraseña en texto plano. La finalidad de esto es poder ver si alguien no autorizado ha intentado entrar al modo administrador.

Las tablas *tm\_page* y *tm\_partial* contienen, respectivamente, los datos de las páginas y de las páginas y de las cabeceras y pies de página. Ambas tienen como clave foránea la clave de la tabla tm\_lang, que solo contiene los registros “es” y “en”, que son los idiomas en los que puede estar la aplicación.

Además, *tm\_page* contiene el nombre del usuario administrador que la creó. Es por ello que la columna *user* es clava foránea que referencia a la tabla *tm\_admin.*

### Diseño de interfaces

En la introducción mostraba la pantalla inicial de la aplicación en la versión de escritorio. A continuación, se puede ver la misma pantalla en versión móvil con el menú desplegado (Fig. 4-4), y con el menú sin desplegar (Fig. 4-5):

. 

Fig. ‑4- Pantalla inicial de la aplicación en la versión con el menú desplegado

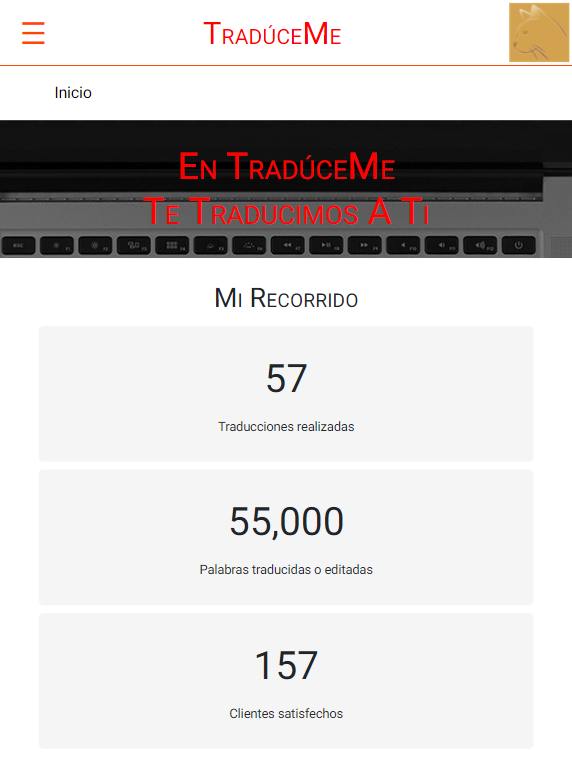


Fig. ‑5- Pantalla inicial de la aplicación en la versión con el menú sin desplegar

El diseño del contenido de las páginas y los pies de página está realizado con Bootstrap y rematado con CSS. Hay diferentes puntos de ruptura que ofrece esta primera herramienta. Son, además los estándares actuales de internet.

Las cabeceras, al ser lo primero que se desarrolló, y sin saber Bootstrap, se fue desarrollado enteramente con HTML y CSS. Sin hacer uso de ninguna otra herramienta.

A pesar de ello, en la última fase del proyecto se realizó algún retoque usando las clases facilitadas por Bootstrap.

Se puede ver su versión inglesa en la figura 4-6:



Fig. ‑6- Cabecera en inglés

En la siguiente figura (Fig. 4-7), el menú en español con un desplegable desplegado. Traducción también tiene su propio desplegable con las páginas de los distintos tipos de traducción.



Fig. ‑7-Cabecera con desplegable

A continuación, detalle del cambio de idioma (Fig. 4-8), las imágenes de ambas banderas han sido realizadas con Inkscape:



Fig. ‑8- Banderas para al cambio de idioma

En la siguiente figura (Fig. 4-9), se puede ver la lista de servicios ofrecidos y el carrusel de opiniones, en la página de inicio en español:

. 

Fig. ‑9- Lista de servicios ofrecidos y carrusel de opiniones

A continuación (Fig. 4-10), el diseño de la página “sobre mí”. La imagen tiene la posición *sticky*, por lo que se mantiene fija navegando por la página s no cabe todo el contenido en la pantalla.

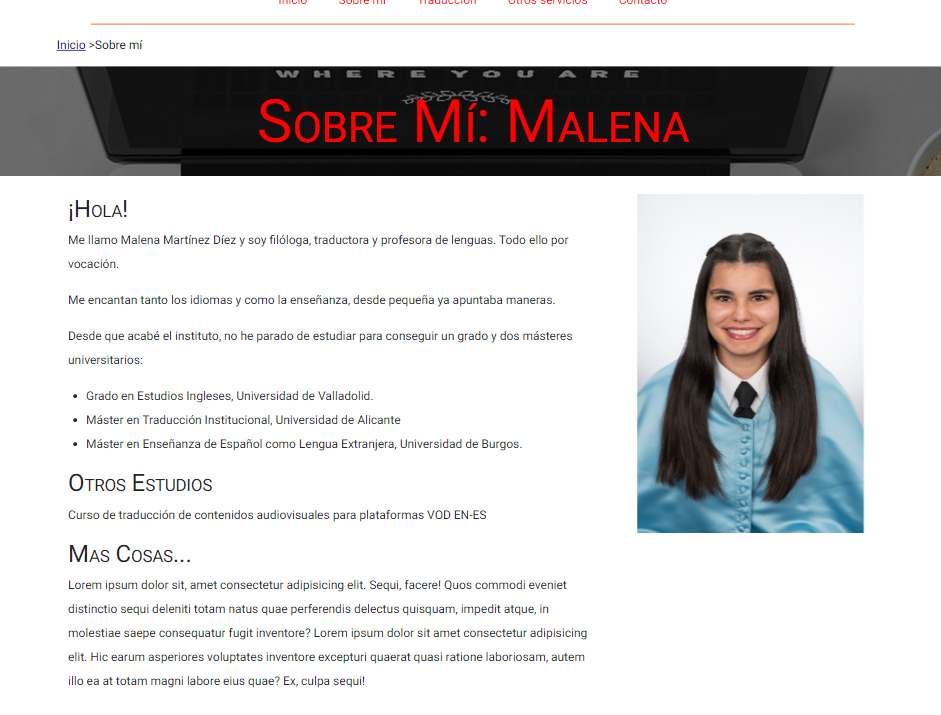


Fig. ‑10- Sobre mí

En la siguiente figura (Fig. 4-11), el diseño de la página “servicios de traducción”. Diseñado con columnas de Bootstrap que ocupan proporciones distintas de la pantalla según el ancho de esta.

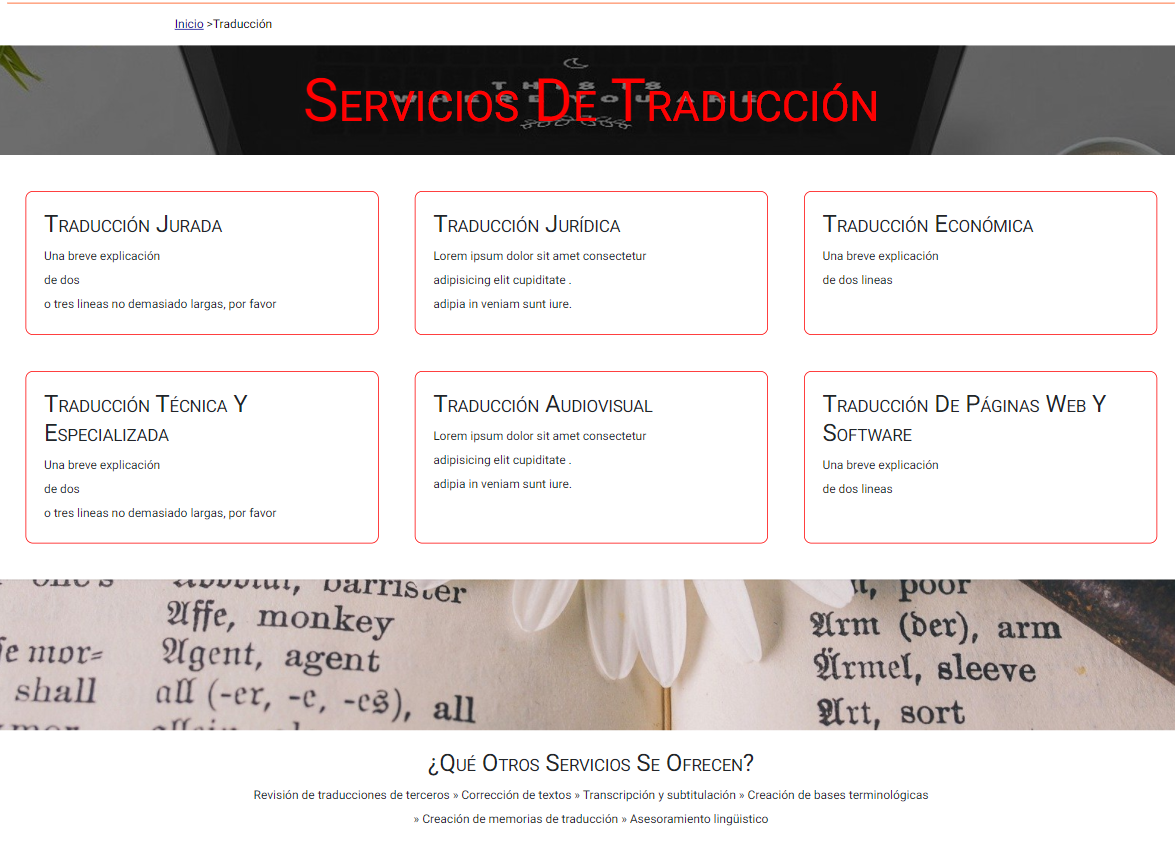


Fig. ‑11-Servicios de traducción

A continuación (Fig. 4-12), el diseño sencillo de la página “contacto” y el píe de página debajo. Dos columnas para pantallas anchas y una solo para pantallas pequeñas.



Fig. ‑12- Contacto

En la siguiente figura (Fig. 4-13), una página modelo para las páginas de traducción y otros servicios. La idea es que la personalice la traductora desde el modo administrador.



Fig. ‑13- Modelo de página

A continuación, el inicio de sesión tras un intento fallido (Fig. 4-14). Es un diseño sencillo, un formulario centrado en la página.

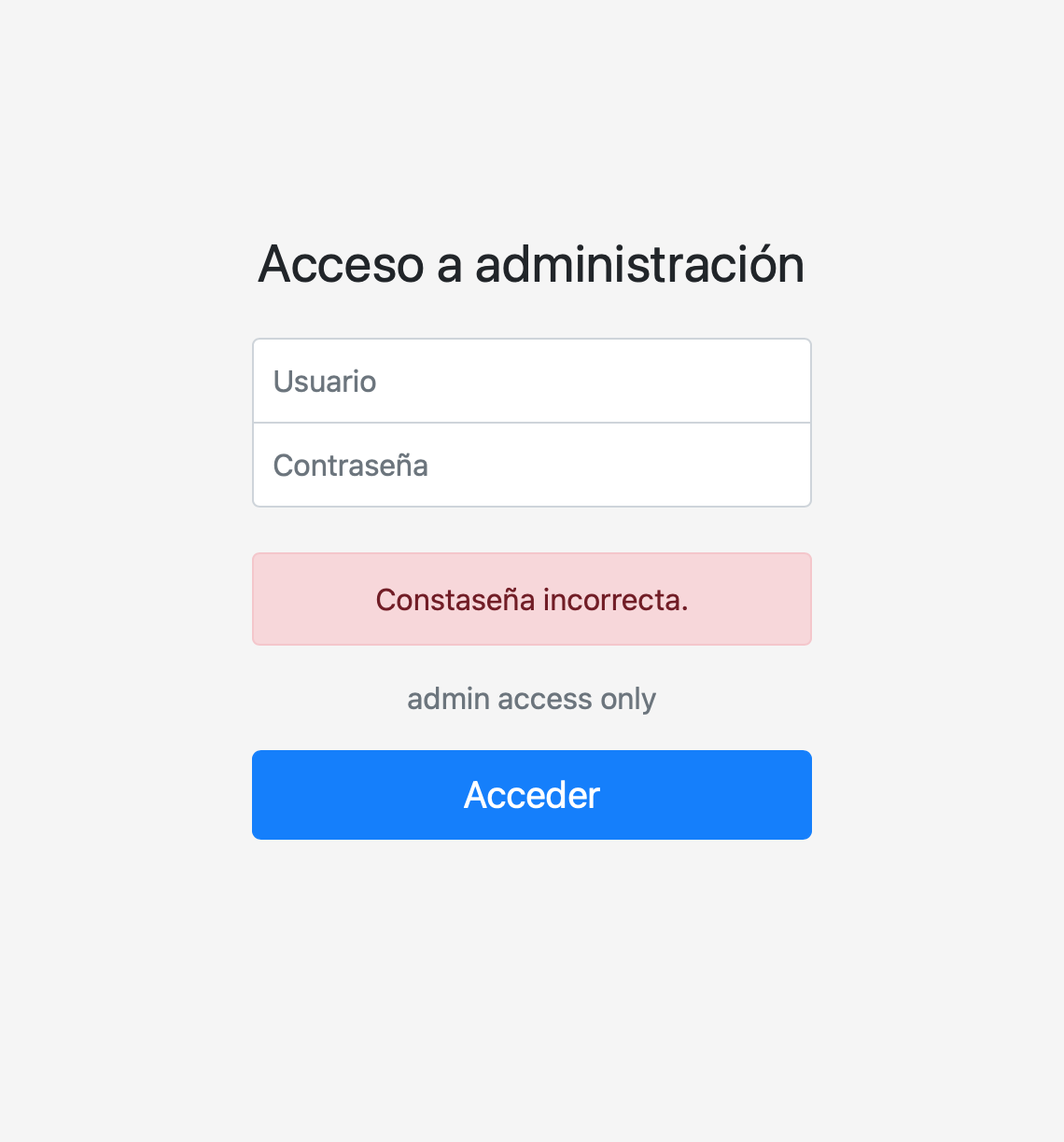


Fig. ‑14- Accesos al modo administrador

En la siguiente figura (Fig. 4-15), la página de edición de páginas después de escoger una página para editar:

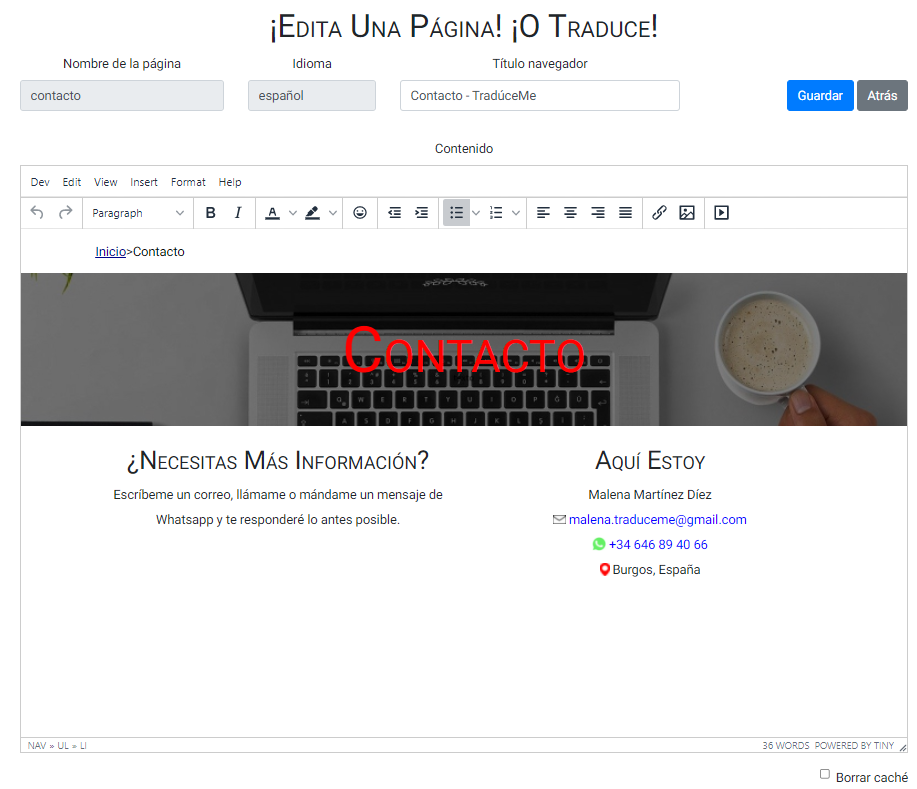


Fig. ‑15- Añadido de páginas: editando página

Debo confesar que Bootstrap ha sido una herramienta esencial para el desarrollo de este proyecto. Antes de descubrirla, estaba convencido de que el diseño de las interfaces me iba a llevar el 70% del tiempo que iba a dedicar al proyecto.

Costó un poco hacerse a ella, pero al poco fue siendo cada vez más fácil e intuitiva.

## Pruebas

En el punto 2 de esta memoria de hablaba de los requisitos funcionales de la aplicación web. Volvemos a tomar el tema, ya que no vale con escribir el código de la aplicación y pensar que funciona. Deben realizarse ciertas comprobaciones para confirmar que la respuesta de la aplicación es la esperada en todo momento.

Ha sido el momento de las pruebas donde se ha comprobado esto y se han corregido los errores y situaciones inesperadas. De esta forma, se ha asegurado el cumplimiento de todos los requisitos que, como decía antes, se mencionaban en el punto 2 de esta misma memoria.

### Pruebas de estilo

Para comprobar que los usuarios vayan a visualizar correctamente la aplicación web, se han hecho pruebas de navegación con los navegadores más comunes: Chrome, Opera, Safari, Mozilla Firefox, Microsoft Edge.

De la misma manera se ha tenido ocasión de comprobarlo con distintas plataformas: ordenadores con MacOS y Windows, una máquina virtual con Ubuntu, móviles Android e iOS, un iPad y un *e-book* Sony Reader de 2015.

Ciertamente ni se esperaba que se viese bien en todos ellos ni así fue. Sin embargo, este era el propósito de las pruebas. Una vez visto el fallo es fácil arreglarlo. Poco a poco, se fue puliendo la aplicación hasta adaptarla a como se tenía que ver.

### Pruebas de funcionalidad

Mis favoritas. Romper la aplicación cada vez que se añade una funcionalidad, es algo que se ha hecho automáticamente, intentando buscar los casos límite y los no controlados. Sin embargo, cierto es que al acabar uno mismo de desarrollarla, uno sabe lo que se espera que ocurra y aunque se intenta hacer pruebas con situaciones inesperadas, uno no se aleja demasiado de lo esperado.

Es inconsciente. Aun así, se encuentran y arreglan fallos. Pero por esto y por asegurar, no han dejado de hacerse pruebas hasta el último momento.

Pasado cierto tiempo de desarrollar una funcionalidad, a uno se le ocurren más formas de *hackearla* y hacer que la aplicación falle de alguna manera. Esto es justo lo que se ha ocurrido. Se han encontrado errores no controlados o situaciones inesperadas.

Por último, y casi lo más importante, queda por hablar de las pruebas de usuario. Donde se hace utilizar a distintos usuarios la aplicación web sin ayuda alguna para que la exploren.

Con esto se pretenden varias cosas. En primer lugar, ver si la aplicación no es intuitiva y necesita algún tipo de guía o la introducción de alguna ayuda durante la ejecución. En segundo lugar, es sabido que sin saber nada de la aplicación hagan cosas que no haría alguien que la conociese bien.

# Conclusiones finales

## Grado de cumplimiento de los requisitos fijados

En cuando a funcionalidades, según se desarrollaban las programadas, han ido apareciendo otras que resultaron ser necesarias o complementarias a las primeras. Por lo que toda la planificación inicial se ha visto modificada en mayor o menor medida a lo largo del desarrollo.

A grandes rasgos, las fases de la planificación inicial se han cumplido. Sin embargo, los tiempos… eso ya es otro cantar. Los tiempos fijados han variado enormemente en absolutamente todas las fases, y hacia arriba.

No es algo de extrañar, puesto que este ha sido el primer proyecto de estas características que he desarrollado. Además, soy quizá demasiado tiquismiquis con algunos detalles y les dedico un tiempo excesivo para ser cosas que apenas se van a notar o ser situaciones que quizá no se vayan a dar. Sin embargo, mi consciencia no me permite omitirlas.

Otro motivo a tener en cuenta a la hora de entender que el tiempo de realización es que se ha aprendido a usar tecnologías nuevas, como Bootstrap, que, si bien ha ahorrado horas de trabajo, ha habido que invertir esas horas en aprendizaje. Muchas veces a base de ensayo y error.

## Propuestas de mejora o ampliaciones futuras

* Sección de cambio de contraseña de administrador.
* Sección para el borrado de páginas de la base de datos
* Posibilidad de ver si las páginas en la base de datos están habilitadas o no en la aplicación.
* Incluir las migas de pan en la tabla tm\_page como una columna en vez de dentro la columna *content*.
* Formulario de contacto que guarde el contenido en la base de datos.
  + Sección en el modo administrador para verlo.
* Añadir un acceso al modo administrador en la interfaz de usuario, solo visible una vez se ha ingresado sesión en el modo administrador.
  + Globo con notificaciones de los envíos de formularios sin leer.

# Guías

## Guía de uso

Comenzamos a viendo las páginas del modo administrador. En primer lugar, la página donde añadir páginas a la interfaz de usuario (Fig. 6‑).

Para añadir una página hace falta introducir un nombre, el idioma en el que estará la página, el título que aparecerá la pestaña del navegador y el contenido de la página haciendo uso del editor visual integrado en la aplicación.

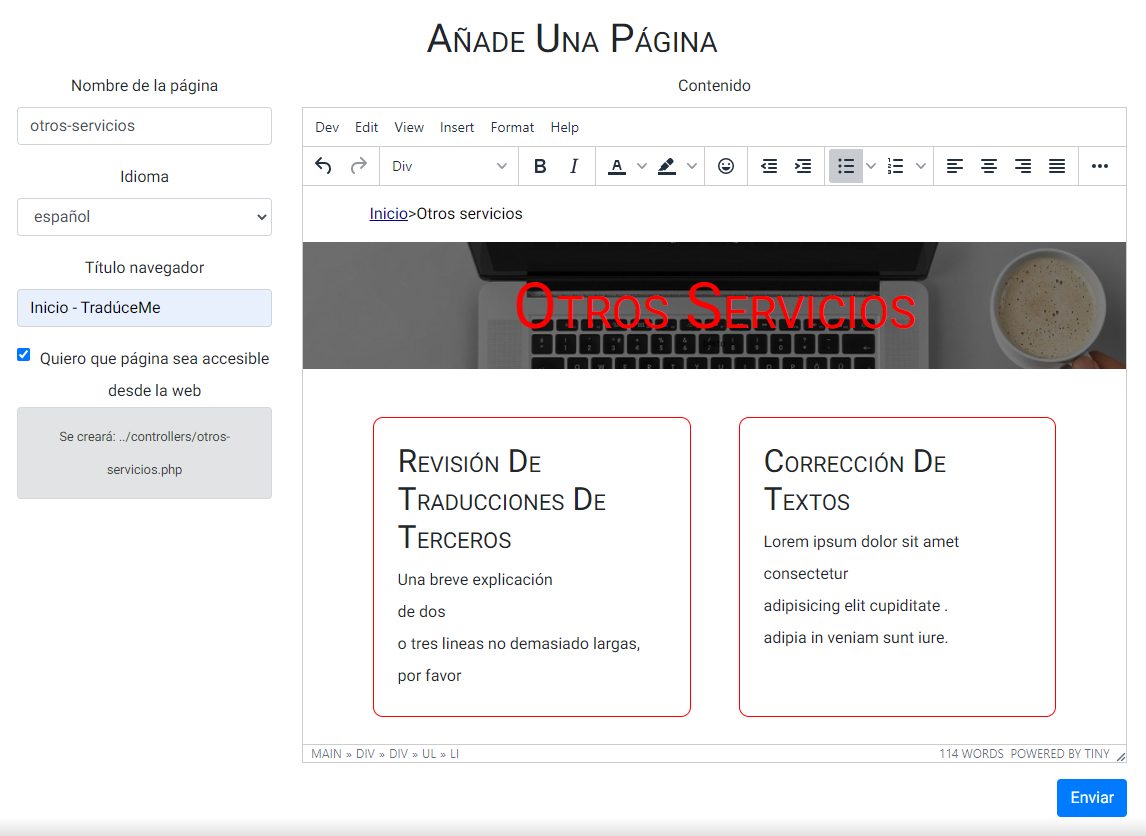


Fig. 6‑- Añadido de páginas

El nombre escogido para la página se utilizará para acceder a ella a través de una URL con la siguiente estructura: *directorio-raíz****/****controllers****/****nombre-de-la-página.php*. Esto si se ha marcado la casilla que indica que se desea posibilitar esto.

En la siguiente figura (Fig. 6-2), el mensaje de éxito al añadir la página. Si ocurre algún error, se recibirá un mensaje de la misma manera que este.

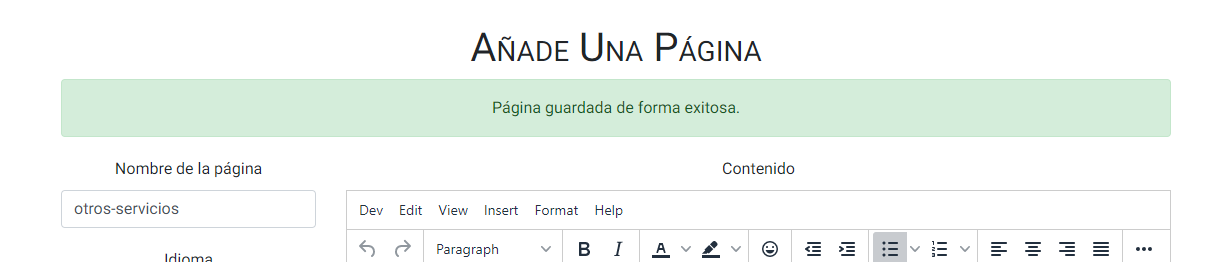


Fig. 6-2- Mensaje de éxito

En la figura de debajo (Fig. 6-3), la página de edición al momento de elegir qué y en qué idioma editar. Si se edita una página se podrá cambiar el contenido y el título del navegador. Si se edita una “parte”, sólo se puede editar el contenido, ya que no tiene títulos para navegador.

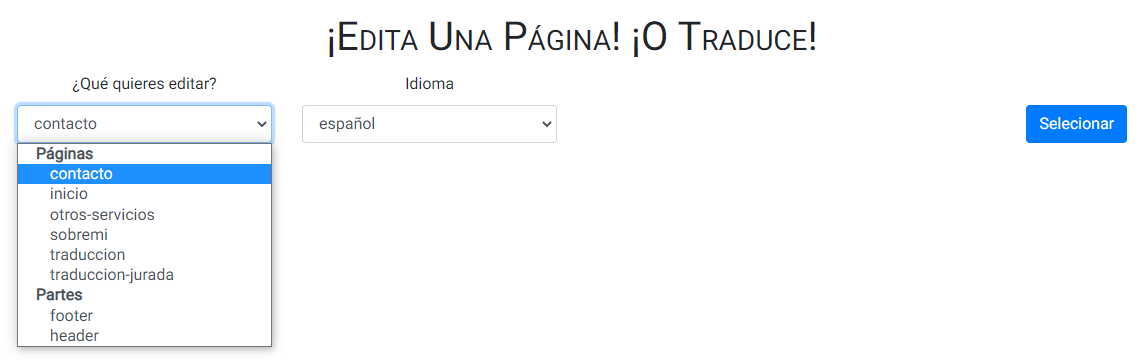


Fig. 6-3- Página de edición: eligiendo

En la siguiente figura (Fig. 6-4), el mensaje de error al editar una página:

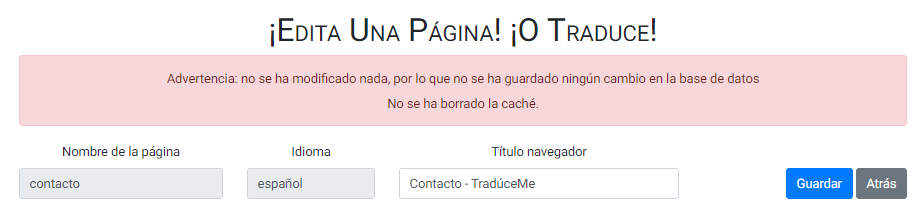


Fig. 6-4- Mensaje de error al editar una pagina

Ahora, la página de edición y comparación (Fig. 6-5). Se elige sólo qué editar, ya que se mostrará lo elegido en ambos idiomas.

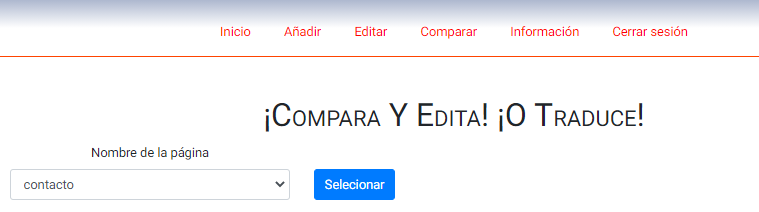


Fig. 6‑5- Página de edición y comparación

En la siguiente figura (Fig. 6-6), mensajes la editar dos páginas. Uno de error y otro no.

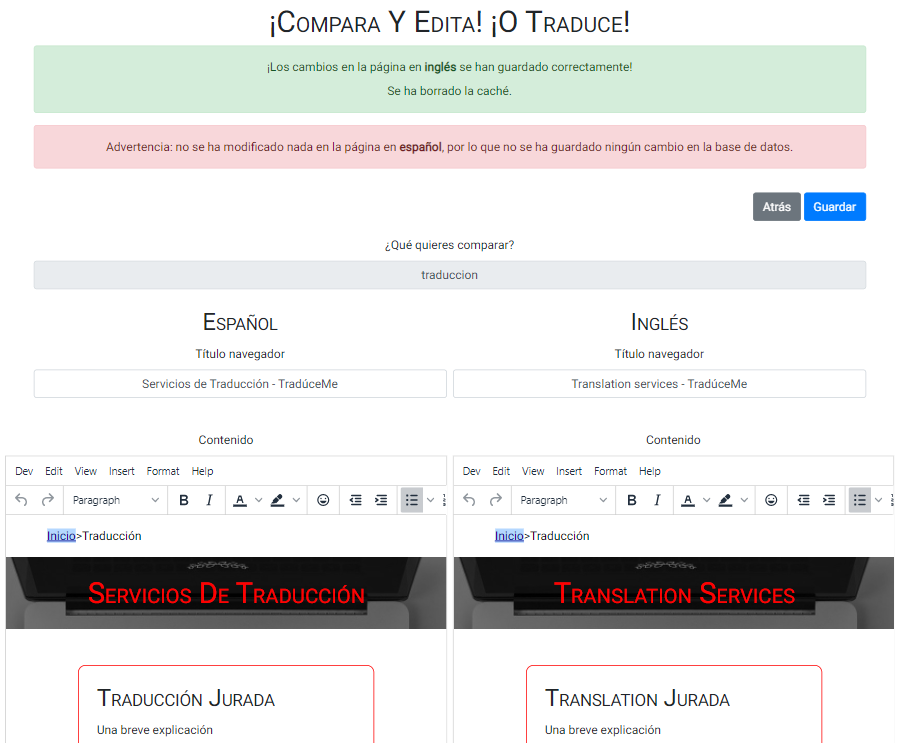


Fig. 6‑6- Mensajes la editar dos páginas

Cabe destacar que tanto al escoger para editar una página en el idioma en el que no está, se muestra el contendido en el idioma en que sí que está. Con esto se espera facilitar la traducción de la web a un simple sustituir texto.

Esto mismo ocurre cuando se quieren comparar dos páginas y solamente esta creada en un idioma.

En la siguiente figura (Fig. 6-7), la vista de la parte superior de página de información, habiendo iniciado sesión con el usuario “mori”:

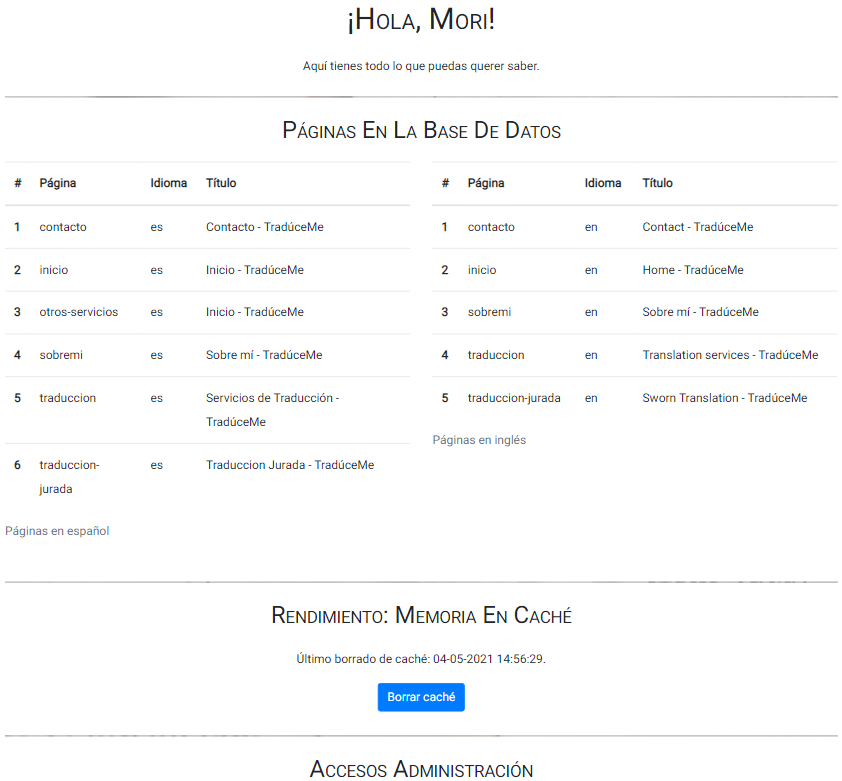


Fig. 6‑7- Parte superior de página de información

Es importante borrar la caché si se edita algo, puesto que la primera vez que se visita una página, esta se guarda en la caché. Los cambios de las ediciones. Sin embargo, se guardan en la base de datos, y al acceder a una página editada y guardada en caché, no se recupera de la base de datos, si no que se utiliza esta.

Es por ello que para ver los cambios puede ser necesario borrar la caché. Se da la opción tanto en la página de información, como cuando se editan las páginas.

Por último, la vista de los accesos al modo administrador (Fig. 6-8). Es la lista del registro de la tabla tm\_access, que como se dijo antes, almacena los intentos de acceso al modo administrador.

No se actualiza en directo, pero casi: al cambiar de página sí que se actualiza si ha habido algún intento de acceso.



Fig. 6‑8- Parte inferior de página de información

Como puede verse, hay contraseñas que se ven y contraseñas que no. Las que no se ven, son de accesos exitosos. Las que se pueden ver, en cambio, son intentos de ingreso fallidos.

## Guía de instalación

El proceso es tan sencillo como mover el contenido de la carpeta TradúceMe a la raíz del servidor —donde van comúnmente los archivos accesibles de las aplicaciones web—. A continuación, crear una base de datos llamada “traduceme” e importar el contenido con el archivo *model/db/sql/traduceme.sql*.

# Referencias

Documentación:

[En línea] - <https://www.w3schools.com> - (Última consulta: 27 de mayo de 2021)

[En línea] - <https://developer.mozilla.org/es/> - (Última consulta: 27 de mayo de 2021)

[En línea] - <https://escss.blogspot.com/2012/10/navicon-tres-lineas-semantico-rwd.html> - (Última consulta: 15 de marzo de 2021)

[En línea] - <https://faq.whatsapp.com/general/chats/how-to-use-click-to-chat/?lang=es> - (Última consulta: 21 de marzo de 2021)

[En línea] - <https://pixabay.com/es/photos/ordenador-portátil-café-el-brazo-1205256/> - (Última consulta: 17 de marzo de 2021)

[En línea]- <https://pixabay.com/es/photos/clave-antigua-flor-nostálgico-5105878/> - (Última consulta: 17 de marzo de 2021)

Referencias:

1. [En línea] - <https://www.apachefriends.org/es/index.html> - (Última consulta: 28 de marzo de 2021) [↑](#endnote-ref-1)
2. [En línea] - <https://code.visualstudio.com> - (Última consulta: 5 de mayo de 2021) [↑](#endnote-ref-2)
3. [En línea] - <https://git-scm.com> - (Última consulta: 13 de marzo de 2021) [↑](#endnote-ref-3)
4. [En línea] - <https://es.wikipedia.org/wiki/GitHub> - (Última consulta: 9 de mayo de 2021) [↑](#endnote-ref-4)
5. [En línea] - <https://es.wikipedia.org/wiki/Inkscape> - (Última consulta: 2 de mayo de 2021) [↑](#endnote-ref-5)
6. [En línea] - <https://developer.mozilla.org/es/docs/Web/Guide/AJAX> - (Última consulta: 20 de abril de 2021) [↑](#endnote-ref-6)
7. [En línea] - <https://jquery.com> - (Última consulta: 25 de abril de 2021) [↑](#endnote-ref-7)
8. [En línea] - <https://getbootstrap.com/docs/5.0/getting-started/introduction/> - (Última consulta: 19 de mayo de 2021) [↑](#endnote-ref-8)
9. [En línea] - <https://httpd.apache.org> - (Última consulta: 25 de mayo de 2021) [↑](#endnote-ref-9)
10. [En línea] - <https://es.wikipedia.org/wiki/MySQL> - (Última consulta: 30 de marzo de 2021) [↑](#endnote-ref-10)
11. [En línea] - <https://www.php.net> - (Última consulta: 27 de mayo de 2021) [↑](#endnote-ref-11)
12. [En línea] - <https://www.tiny.cloud/docs/configure/> - (Última consulta: 19 de abril de 2021) [↑](#endnote-ref-12)
13. [En línea]- <https://github.com/tinymce/tinymce> - (Última consulta: 17 de marzo de 2021)

    [↑](#endnote-ref-13)