

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

[Contenido 2](#_Toc72686644)

[Índice de figuras 3](#_Toc72686645)

[Índice de tablas 4](#_Toc72686646)

[1. Introducción 5](#_Toc72686647)

[1.1. Descripción 5](#_Toc72686648)

[1.2. Justificación 8](#_Toc72686649)

[2. Planificación 8](#_Toc72686650)

[2.1. Requisitos 8](#_Toc72686651)

[2.1.1. Requisitos funcionales 8](#_Toc72686652)

[2.1.2. Requisitos no funcionales 8](#_Toc72686653)

[2.2. Recursos 9](#_Toc72686654)

[2.2.1. Recursos hardware 9](#_Toc72686655)

[2.2.2. Recursos software 9](#_Toc72686656)

[2.3. Planificación temporal 9](#_Toc72686657)

[2.4. Planificación económica 9](#_Toc72686658)

[3. Tecnologías 11](#_Toc72686659)

[4. Desarrollo y secuenciación temporal 12](#_Toc72686660)

[4.1. Diseño 12](#_Toc72686661)

[4.1.1. Diagrama de aplicación 12](#_Toc72686662)

[4.1.2. Diagrama E/R (entidad/relación) y esquema de BBDD 12](#_Toc72686663)

[4.1.3. Diseño de interfaces 12](#_Toc72686664)

[4.2. Pruebas 12](#_Toc72686665)

[5. Conclusiones finales 13](#_Toc72686666)

[5.1. Grado de cumplimiento de los requisitos fijados 13](#_Toc72686667)

[5.2. Propuestas de mejora o ampliaciones futuras 13](#_Toc72686668)

[6. Guías 14](#_Toc72686669)

[7. Referencias 15](#_Toc72686670)

# Índice de figuras

[Fig. 1‑1-Pantalla inicial de la aplicación 6](#_Toc477346553)

[Fig. 4‑1-Alta de clientes 10](#_Toc477346554)

# Índice de tablas

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

[Tabla 2‑2- Presupuesto del proyecto 8](#_Toc477346556)

# 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.

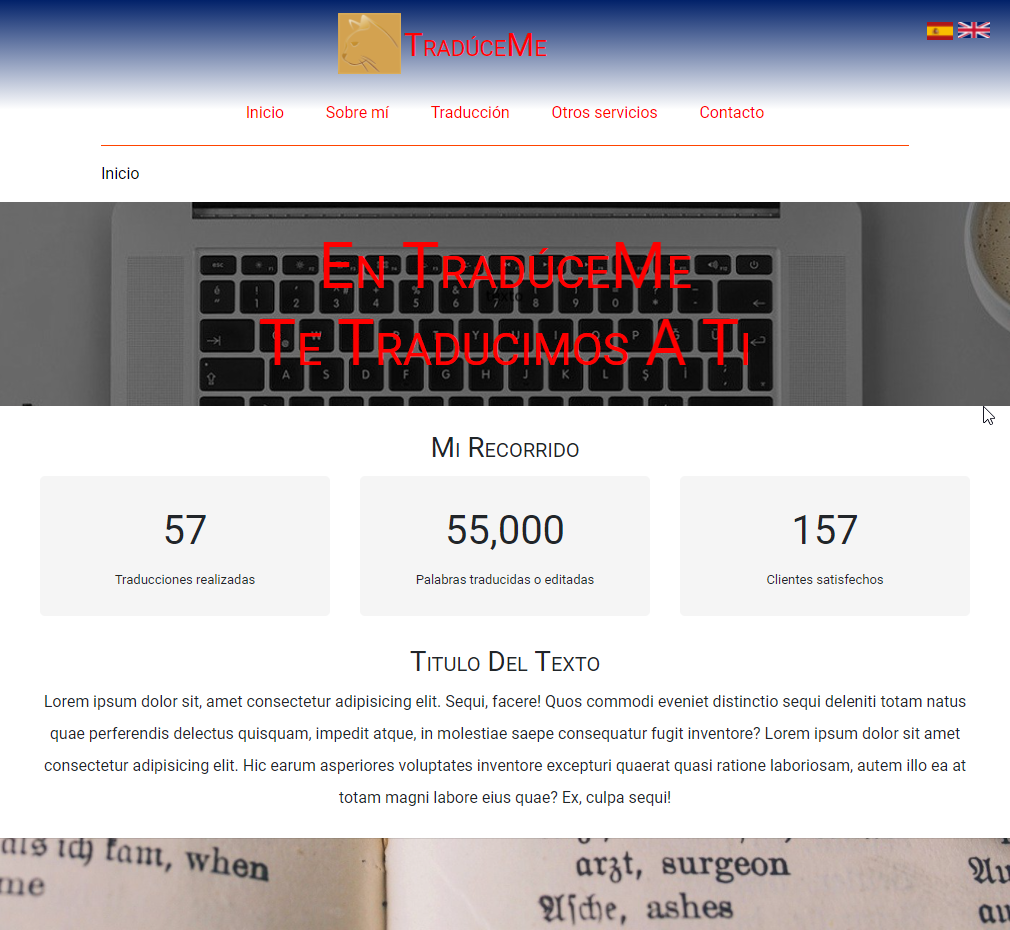


Fig. ‑-Pantalla inicial de la aplicación

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.

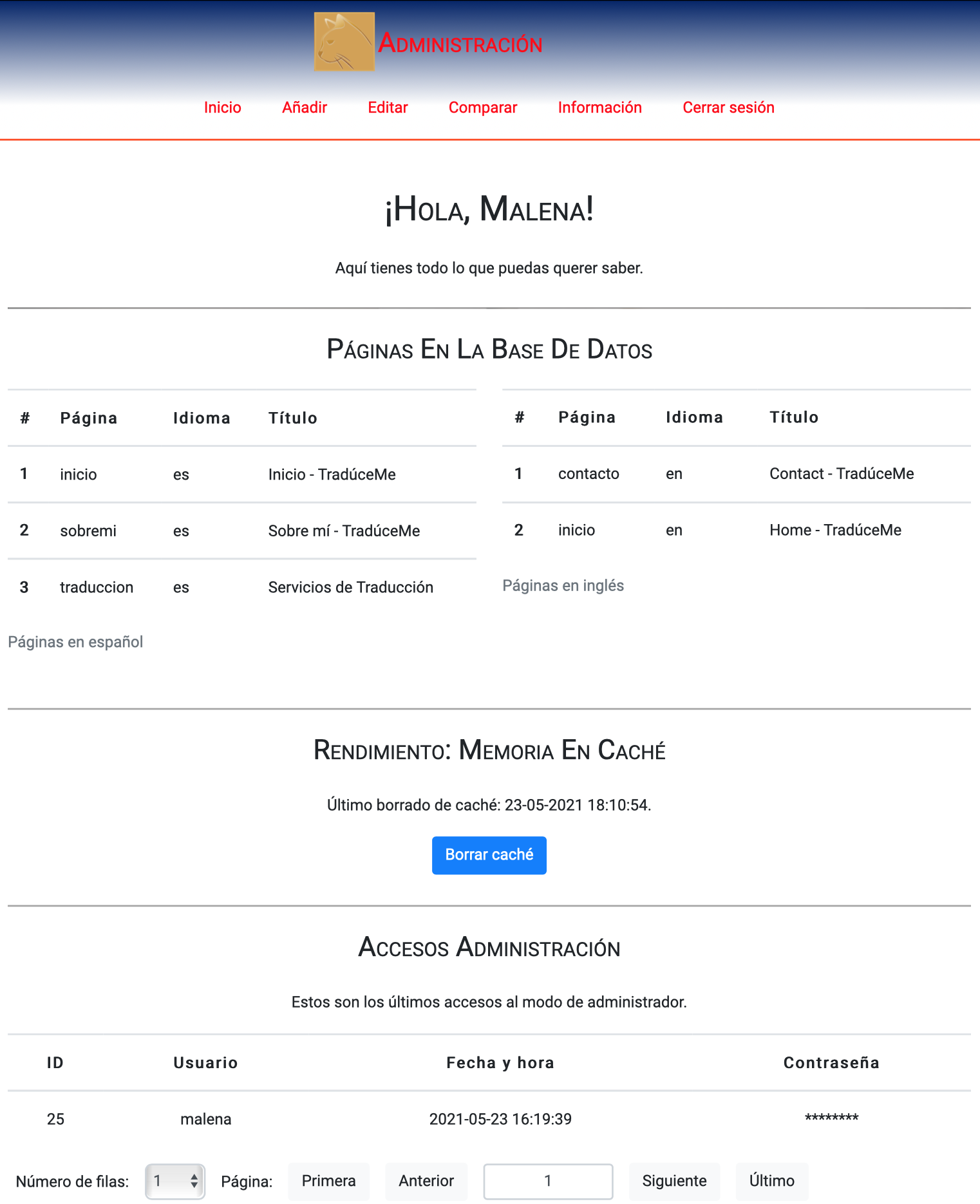


Fig. ‑-Pantalla de la sección de información del modo administrador

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 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.

Esto último, hace no demasiados años, era algo que podría considerarse incluso extravagante para personas particulares. Sin embargo, cada día es más común e incluso necesario. Pues este espacio en internet, en primer lugar, permite que te encuentren fácilmente y sin conocerte de antemano, y, en segundo, se traduce —y nunca mejor dicho— en que el cliente tendrá un mayor sentimiento de confianza a la hora de decidir a quién contratar los servicios.

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 capaz de ejecutar XAMPP.

### Recursos software

Lista de recursos software

* XAMPP
* Visual Studio Code
* Git, GitHub y GitHub Desktop
* Navegadores:
  + Chrome
  + Opera
  + Safari
  + Mozilla Firefox
  + Microsoft Edge
* Inkscape

## Planificación temporal

La fase inicial no es otra que la del **prototipadado 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.

Ultima 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** |
| Prototipadado de la aplicación web | 2 |
| Diseño y relleno de la interfaz de usuario | 8 |
| Generación dinámica de la web | 8 |
| Desarrollo modo administrador | 8 |
| Caché y seguridad | 4 |
| Detalles y arreglos | ≥ 0 |
| **TOTAL HORAS** | ≥ 30 |

## 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 esta respaldado por toda una comunidad.

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

# Tecnologías

## HTML

## CSS

## Bootstrap

## JavaScript

## JQuery

## JSON

## PHP

## MySQL

## Inkscape

## 

Tecnologías que se van a utilizar. Breve reseña y referencias externas.

Todas las tecnologías empleadas llevarán una descripción, más o menos extensa, en función del tratamiento que se les haya dado a lo largo del ciclo.

# Desarrollo y secuenciación temporal

## Diseño

### Diagrama de aplicación

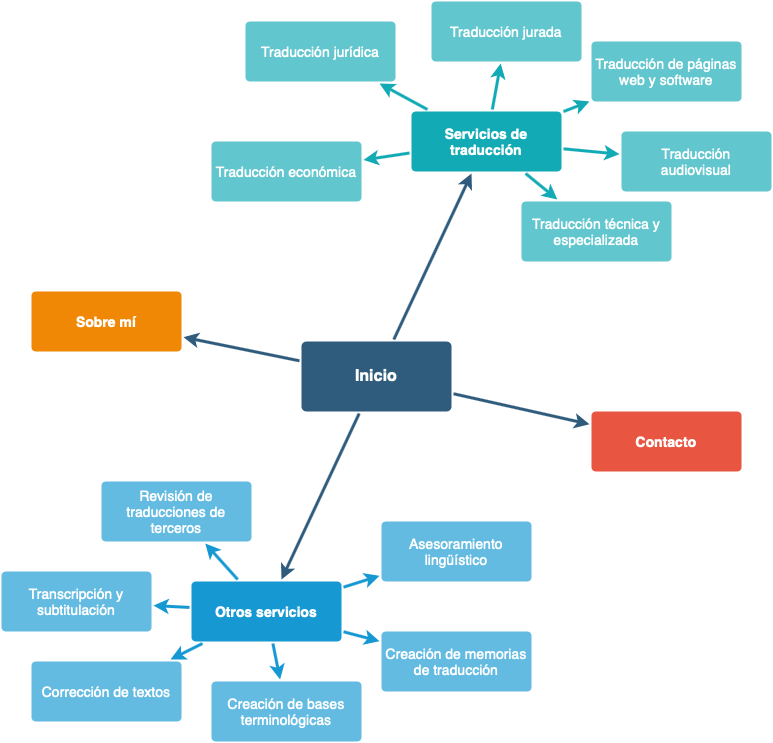


Fig. 4‑1-Diagrama de la interfaz de usuario

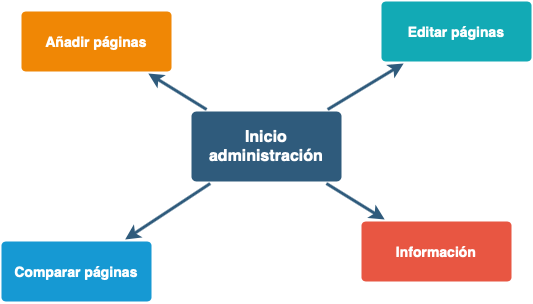


Fig. 4‑2-Diagrama del modo administrador

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

### Diseño de interfaces

En la siguiente figura (Fig. 4-1) ….

. 

Fig. ‑-Alta de clientes

## Pruebas

aaaa

# Conclusiones finales

## Grado de cumplimiento de los requisitos fijados

*Analizar el grado de cumplimiento de los requisitos fijados y análisis de las desviaciones.*

## Propuestas de mejora o ampliaciones futuras

* *aaaa*
* *aaaa*

# Guías

*Si procede*

# Referencias

|  |  |
| --- | --- |
| [1] | [En línea]. Available: https://es.wikipedia.org/wiki/Monta%C3%B1as\_Rocosas. [Último acceso: 15 Marzo 2020]. |
| [2] | [En línea]. Available: https://www.w3.org/TR/html5/. [Último acceso: 15 03 2020]. |

*Incluirá toda la documentación consultada: libros, apuntes, páginas web, etc.*

*NO se incluirán documentaciones exhaustivas sino que se recurrirá de forma habitual a usar buenas referencias.*

*En el último apartado, bibliografía, se incluirá un listado de las referencias indicando autor, título, fecha y en su caso dirección web.*