MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Web

Autor: Asier Martín González

Tutor: Sergio Malo Molina

**t\_GIRA**

2S2021

Enlace Google Drive: https://drive.google.com/drive/folders/1aM-2SJLBY59AcMzc7oYTnEJRDNMPalwn?usp=sharing



**INDICE**

1-. INTRODUCCION

2.- MOTIVACION

3.- ABSTRACT

4.- OBJETIVOS PROPUESTOS (GENERALES Y ESPECÍFICOS)

5.- METODOLOGIA UTILIZADA

6.- TECNOLOGIAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO

7.- ESTIMACION DE RECURSOS Y PLANIFICACION

8.- ANALISIS

9.- DISEÑO

10.- DESPLIEGUE Y PRUEBAS

11.- CONCLUSIONES

12.- VIAS FUTURAS

13.- GLOSARIO

14.- BIBLIOGRAFIA / WEBGRAFIA

15.- ANEXOS

**1.- INTRODUCCION**

La idea surge de la necesidad de controlar las tareas que se asignan a los trabajadores y poder prescindir de papel.

En la empresa en la que trabajo actualmente, disponemos de un sistema para guardar las tareas que vamos realizando, pero a día de hoy, no disponemos de ningún sistema fiable para adjudicar tareas a los trabajadores de forma clara y fiable. Utilizamos Skype para adjudicar las tareas a los trabajadores, pero no es un método efectivo ni práctico, ya que no quedan todas reflejadas de forma clara y en un lugar accesible de manera rápida.

Personalmente comencé a utilizar “ASANA”, que es una web donde puedes crear tareas de forma rápida y que dispone de una interfaz gráfica fácil de entender. Por desgracia, los responsables de la empresa no quieren que se guarden datos de usuarios fuera de los servidores propios.

Fue aquí cuando nació la idea de crear este sistema, denominado t\_GIRA.

En la empresa se utiliza la inicial del proyecto para indicar de que se trata la aplicación (en este caso la “t” es de tareas) y la denominación GIRA, que son las siglas de la empresa. De ahí el nombre de la aplicación que aquí se presenta.

El administrador podrá crear, modificar y eliminar tareas que luego podrá adjudicar a los empleados de una manera muy sencilla.

Imagen 1 - Título: Logo Gira -

Fuente: gira.es

Además, podrá ver cuando estas tareas estén concluidas. También podrá crear, modificar y eliminar clientes. De la misma manera, también podrá crear, modificar y eliminar usuarios(empleados).

El usuario(empleado) dispone de un menú en el que aparece una tabla con las tareas que tiene asignadas. También dispone de un botón que le da acceso a un listado de clientes. De la misma manera, también dispone de un botón que le da acceso a un listado con los datos de los usuarios. Cuando pulse en el botón comenzar, se desplegarán los datos del cliente y los pasos de la tarea. Estos pasos se podrán ir marcando según se vayan realizando para llevar un control.

Una vez terminada la tarea, el usuario puede hacer clic en el botón Terminar Tarea y automáticamente la tarea desaparecerá de su lista de tareas pendientes y al administrador le aparecerá como tarea Realizada.

Si el administrador lo desea, puede eliminar de la tabla las tareas que ya se han realizado.

**2.- MOTIVACION**

Este proyecto nace con la idea de buscar una mejor organización de las tareas en el trabajo. La idea principal es facilitar el trabajo del administrador a la hora de asignar las tareas a los empleados y que esto se haga de forma intuitiva y práctica.

En un futuro se intentará dotar a la aplicación de ciertas mejoras, tales como una mayor interacción por parte de los empleados hacia el administrador de la aplicación para llevar a cabo un mejor seguimiento de las tareas asignadas.

Con esta aplicación se pretende llevar a uso todo lo aprendido a lo largo de estos dos años de formación y a la vez aprovecharlo para intentar hacer un aplicativo que realmente sea útil.

A la vez se aprovecha para practicar algún lenguaje de programación buscando algún reto a la hora de desarrollar la aplicación.

**3.- ABSTRACT**

In this project we try to carry out what was learned during the years of training by creating an application capable of being used in a company.

The highlight of this application is the ease with which the administrator can assign tasks to the workers. In addition, it is extremely easy to create, modify and delete tasks, users and clients.This can be done from very clear and intuitive minuses.

For their part, users have a menu where they are shown the tasks they have to do. These tasks, once completed, can be marked as completed so that the administrator can see it reflected in his menu and can remove them from his to-do list if he wishes. Once the user selects a task to carry out, the most important steps for carrying it out will be shown, as well as the data of the client on which the task is carried out in case it is necessary to contact him. All forms are equipped with systems to avoid the injection of SQL code. The administrator also has a menu from which to customize some application design parameters such as the background colors and the logo.

**4.- OBJETIVOS PROPUESTOS (GENERALES Y ESPECIFICOS)**

**OBJETIVOS GENERALES**

El objetivo de este proyecto denominado t\_GIRA, es la creación de una aplicación web, capaz de asignar tareas por parte del administrador al resto de los usuarios, usando para ello lo aprendido durante el Grado Superior de Desarrollo de Aplicaciones Web.

**OBJETIVOS ESPECIFICOS**

Practicar e investigar más sobre los lenguajes de programación y lenguaje de etiquetas:

-HTML

-CSS

-JavaScript

-Jquery

-PHP

-Bootstrap4

Practicar con el uso de algún framework de estilos y aprender más sobre el mismo:

-Bootstrap4

Practicar con el uso de las BBDD (bases de datos) con:

-MySQL

-PHPMyAdmin

Practicar y aprender nuevas funcionalidades sobre el IDE:

-Visual Studio Code

Aprender a llevar un control y gestión sobre el avance de los proyectos :

-GitHub

Compartir trabajos en la nube:

-GoogleDrive

Practicar el diseño de diagramas para los Casos de Uso y Modelos de Datos:

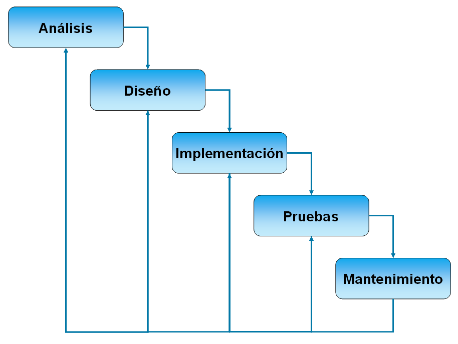
-UML

Realizar buenas prácticas de programación.

Diseño responsivo.

**5.- METODOLOGIA UTILIZADA**

Para la realización del proyecto, en un principio se optó por el Modelo de Desarrollo en Cascada. Al poco de comenzar el proyecto se vio que no era la metodología correcta, ya que a veces era necesario realizar cambios en alguno de los pasos que ya se habían realizado. Por tanto, se descartó la utilización del Modelo en Cascada y se optó por el Modelo en Cascada con Realimentación.

Aun así, el proyecto se dividió en varios subproyectos, para llevar un mejor control. Login, Tareas, Administrador, Usuarios, Clientes y Configuración. De esta manera, cada uno de los anteriores pasó por las etapas de Análisis, Diseño, Codificación, Pruebas y Mantenimiento. A su vez, todos los subproyectos pasaron por esas etapas en conjunto.

Con este modelo, se pudo ir trabajando en el proyecto dividiéndolo en pequeños proyectos. De esta manera, pongamos el ejemplo del Login, se pudo analizar, diseñar, programar y probar de manera independiente.

Imagen 2 - Título: Modelo en cascada con realimentación –

Fuente: Blogspot.com

Estas fueron las fases:

-Análisis, planificación de requisitos y casos de usos funcionales y no funcionales.

-Diseño de los menús.

-Programación de cada bloque o subproyecto.

-Pruebas de cada bloque o subproyecto y pruebas en conjunto.

-Mantenimiento. Se han estudiado las maneras de mantener y evolucionar la aplicación.

**6.- TECNOLOGIAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO**

* Para el SO (sistema operativo) se ha optado por la utilización de Windows 10 Profesional. Es el SO al que se ha tenido acceso tanto en el hogar como en el puesto de trabajo.
* El equipo que se ha utilizado para el desarrollo se compone de un ordenador portátil de la marca LG, modelo gran (gama alta), 16 GB de RAM, procesador intel core i5 9ª generación y memoria SSD de 1TB Samsung Evo. Ocasionalmente un ordenador de escritorio HP Prodesk 600, 32 GB de RAM, procesador Intel i7 10ª generación y memoria SSD de 500 GB Samnsung. También se han utilizado 3 monitores de 15, 25 y 32 pulgadas.
* El IDE utilizado ha sido Visual Studio Code, por estar altamente familiarizado con él y disponer de multitud de extensiones gratuitas.
* Como navegadores para realizar las pruebas de diseño, se han utilizado el Google Chrome, Firefox Developer Edition, Microsoft Edge y Opera por ser estos los más utilizados en la actualidad.
* Como base de datos se ha utilizado MySQL y el gestor PHPMyAdmin, ya que dispone de una interfaz gráfica muy potente e intuitiva, lo que facilita el trabajo.

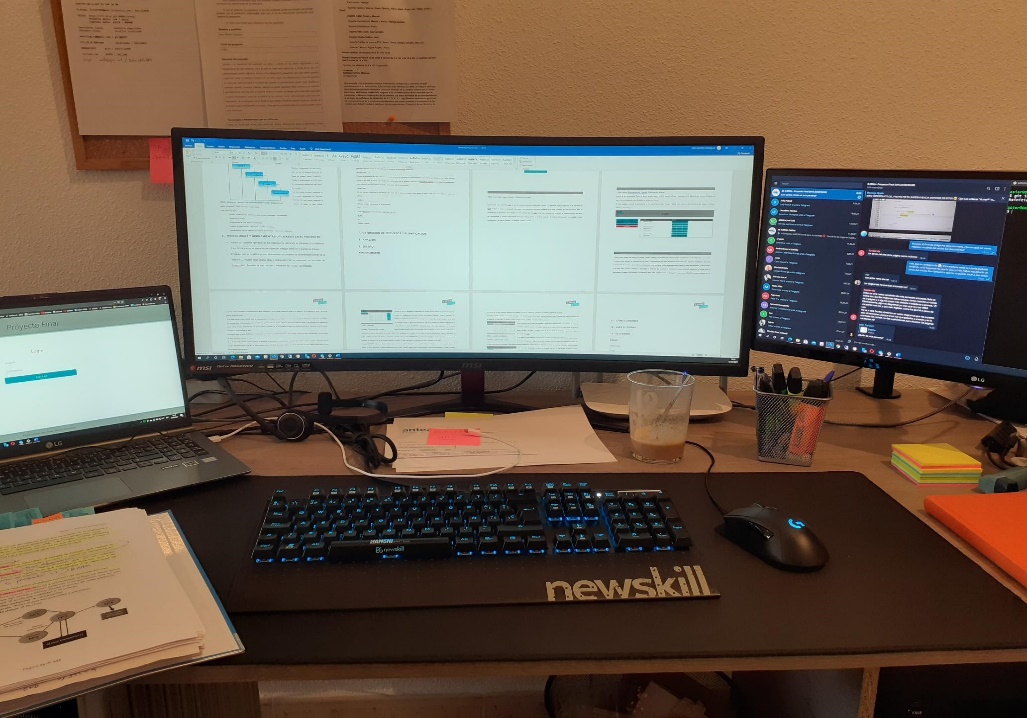


Imagen 3 - Título: Equipo hogar - Fuente: Elaboración propia

* Para realizar los distintos diagramas se ha utilizado el software gratuito draw.io, ya que ha sido utilizado con anterioridad y los resultados fueron satisfactorios.
* Para el diagrama de Gantt se ha utilizado una hoja de Excel, ya que no se disponía de tiempo suficiente para investigar sobre las aplicaciones disponibles para realizar tal tarea.
* Para llevar un control de las tareas, se ha utilizado el software online ASANA, fácil de usar y gratis para las funcionalidades que se han usado.
* Lenguajes y frameworks para el Frontend:

-HTML

-CSS

-JavaScript

-JQuery

-Bootstrap4

Todos ellos se han elegido por estar familiarizado con su uso, salvo Bootstrap, que se ha elegido debido a la fuerte demanda actual.

* Lenguaje para el Backend:

-PHP

* Para el control de versiones se ha utilizado Git. El proyecto está subido a GitHub. Enlace al proyecto:

<https://github.com/asiermgonzalez/t_gira>

* Para el diseño de clases, diagramas, etc. se ha utilizado Lucidchart, por ser gratuito y de fácil uso.
* Para compartir el proyecto con el profesorado se ha utilizado GoogleDrive.

**7.- ESTIMACION DE RECURSOS Y PLANIFICACION**

Para llevar a cabo un control de la planificación, se ha creado un diagrama de Gantt muy sencillo. Puesto que el proyecto se ve sometido a cambios durante su desarrollo, es difícil llegar a cumplir todos los objetivos marcados en su plazo determinado. No obstante, se ha intentado acercarse lo máximo posible a los plazos. Las franjas están separadas por días.

El diagrama está realizado teniendo en cuenta la fecha de entrega ordinaria.

En cada actividad, se muestra el tiempo real empleado y, justo debajo de cada una de ellas, el tiempo que se había estimado antes de comenzar la aplicación.

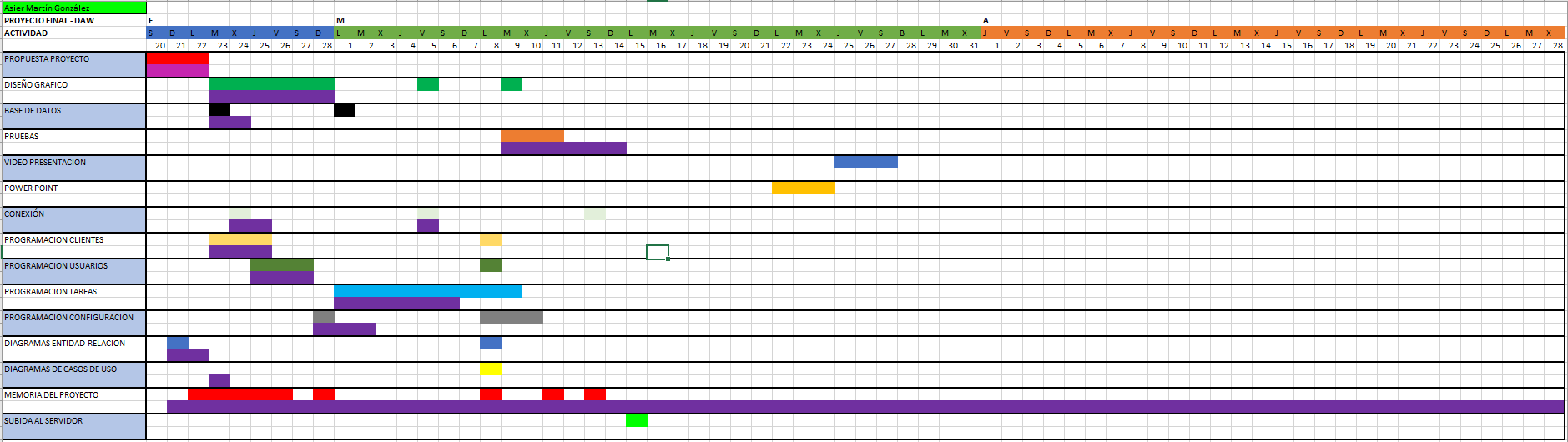


Imagen 4 - Título: Diagrama de Gantt - Fuente: Elaboración propia

Debido al horario laboral, las horas destinadas al proyecto han sido prácticamente siempre las misas: de las 19:00 a las 22:30h.

El tiempo total aproximado ha sido de unas 300 horas.

**8.- ANALISIS**

**REQUISITOS FUNCIONALES**

RF-01: El administrador o los usuarios deben logearse para poder acceder a sus respectivos menús.

RF-02: El administrador debe poder crear, modificar, eliminar y asignar tareas a los usuarios.

RF-03: El administrador debe poder crear, modificar y eliminar usuarios.

RF-04: El administrador debe poder crear, modificar y eliminar clientes.

RF-05: El administrador debe poder modificar algunos aspectos gráficos del diseño de la aplicación.

RF-06: El administrador debe poder visualizar las tareas que has realizado los usuarios o estén pendientes de realizar.

RF-07: El administrador debe poder eliminar tareas asignadas a los usuarios tanto si ya se han realizado como si aún no se han realizado.

RF-08: El usuario debe poder desplegar las tareas que le han sido asignadas.

RF-09: El usuario debe poder visualizar el listado de clientes.

RF-10: El usuario debe poder visualizar el listado de los usuarios.

RF-11: El usuario debe poder marcar como finalizada una tarea asignada.

RF-12: El listado de los clientes será por orden alfabético del nombre.

RF-13: El listado de los usuarios será por orden alfabético del nombre comercial.

RF-14: Tanto el administrador como los usuarios deben de poder cerrar sesión desde cualquier menú en el que se encuentre.

**REQUISITOS NO FUNCIONALES**

RNF-01: Los lenguajes de programación del fronted deben ser JavaScript y JQuery.

RNF-02: El framework para el diseño será Bootstrap4.

RNF-03: El lenguaje del backend debe ser PHP.

RNF-04: El acceso a los distintos menús deberá ser instantáneo, no podrá demorarse más de 1 segundo.

RNF-05: Las tablas y formularios deben ser de diseño claro.

RNF-06: La paleta de colores empleada debe ser la misma para todas las páginas.

RNF-07: En lo posible, se utilizarán mayúsculas para facilitar la lectura.

RNF-08: Separar el código en distintos archivos para un fácil mantenimiento.

RNF-09: Utilizar comentarios en el código.

RNF-10: Primar la funcionalidad por encima del diseño.

RNF-11: Usar Git para controlar las versiones.

**DIAGRAMA DE CLASES**

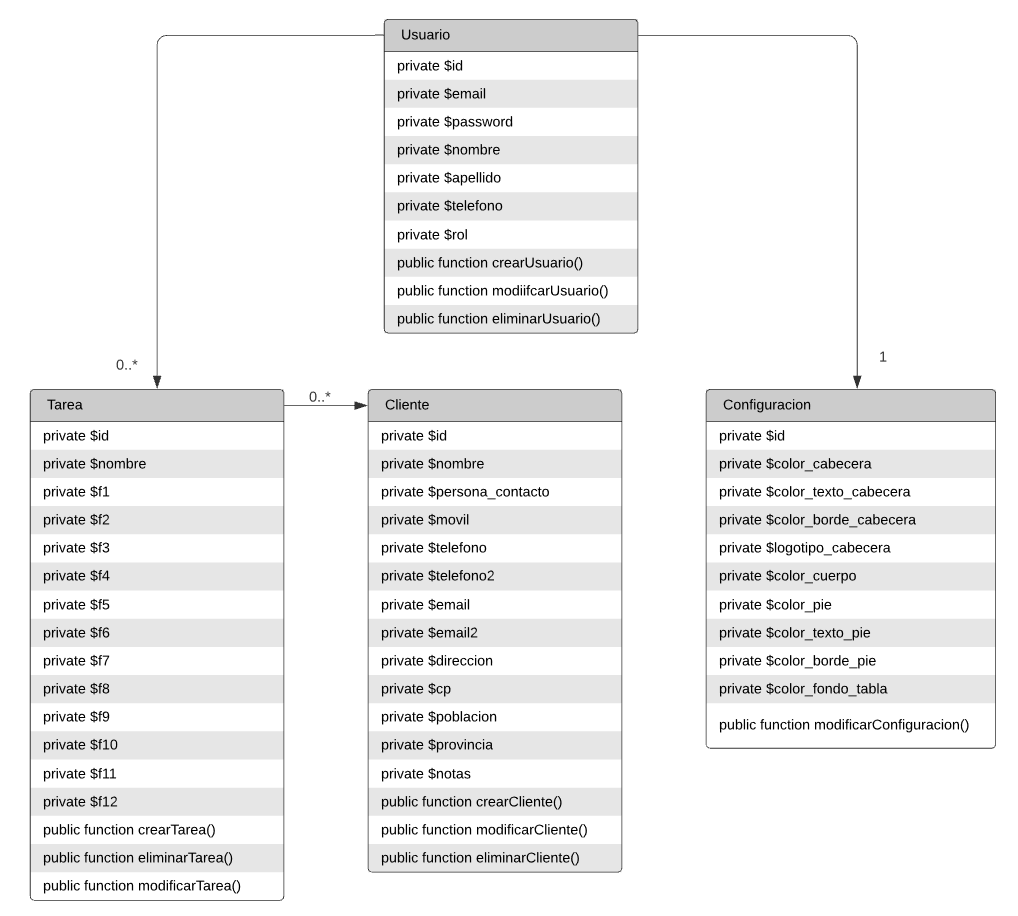
****

Imagen 5 - Título: Diagrama de Clases - Fuente: Elaboración propia

**CASOS DE USO**

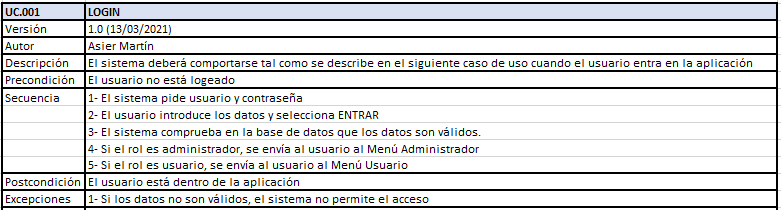
****

Imagen 6 - Título: UC.001 - Fuente: Elaboración propia

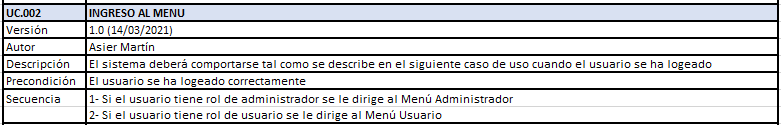
****

Imagen 7 - Título: UC.002 - Fuente: Elaboración propia

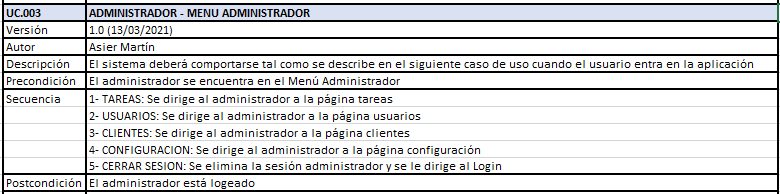
****

Imagen 8 - Título: UC.003 - Fuente: Elaboración propia

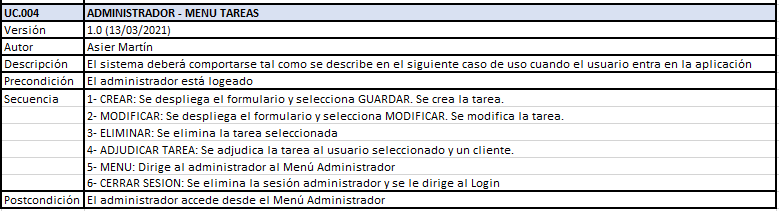
****

Imagen 9 - Título: UC.004 - Fuente: Elaboración propia

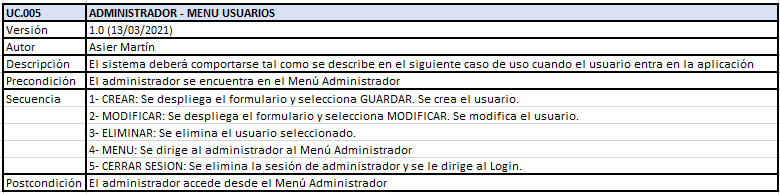
****

Imagen 10 - Título: UC.005 - Fuente: Elaboración propia

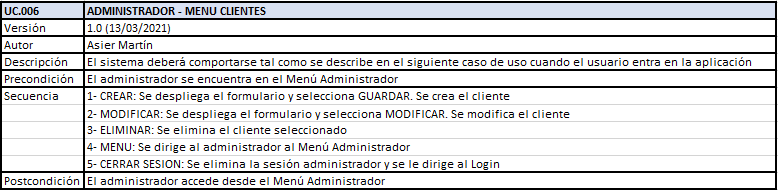
****

Imagen 11 - Título: UC.006 - Fuente: Elaboración propia

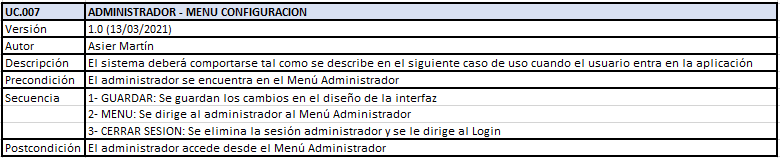
****

Imagen 12 - Título: UC.007 - Fuente: Elaboración propia

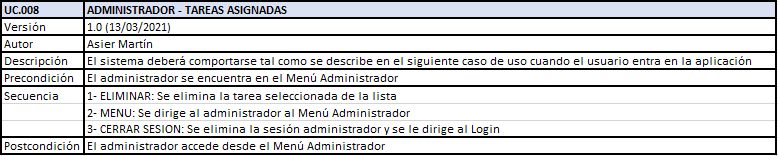
****

Imagen 13 - Título: UC.008 - Fuente: Elaboración propia

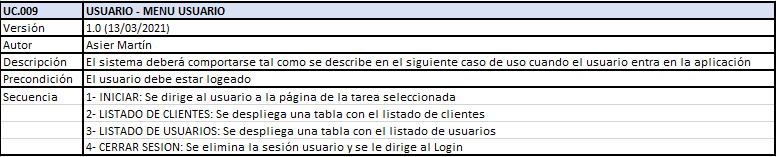
****

Imagen 14 - Título: UC.009 - Fuente: Elaboración propia

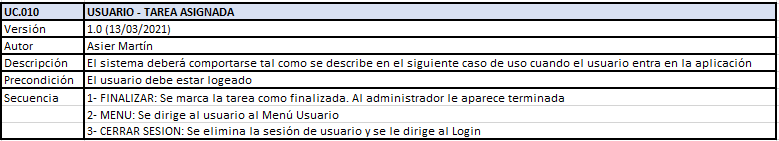
****

Imagen 15 - Título: UC.010 - Fuente: Elaboración propia

**DIAGRAMAS DE CASOS DE USO**

**Proceso de Login**

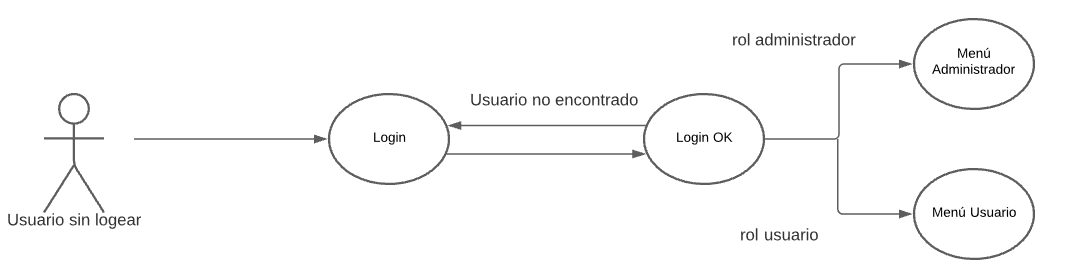
****

Imagen 16 - Título: CU Proceso de login - Fuente: Elaboración propia

**Opciones del Administrador**

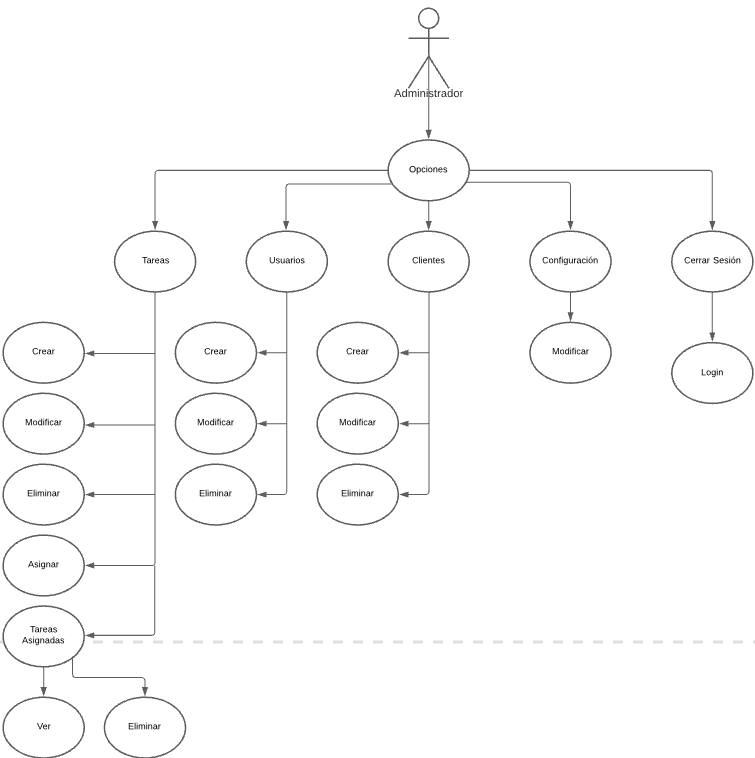
****

Imagen 17 - Título: Opciones del Administrador - Fuente: Elaboración propia

**Opciones del Usuario**

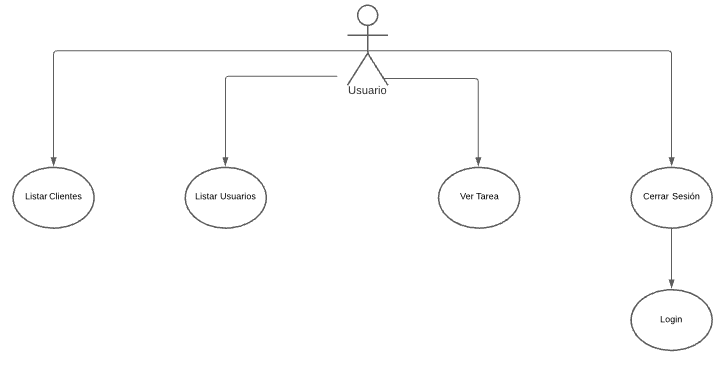
****

Imagen 18 - Título: Opciones del Usuario - Fuente: Elaboración propia

**Tareas Asignadas**

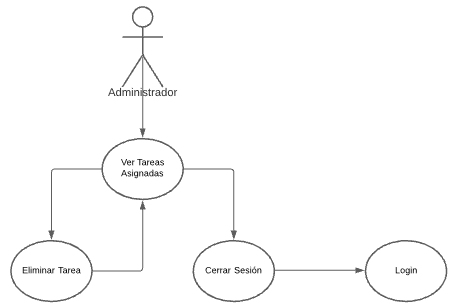
****

Imagen 19 - Título: Tareas Asignadas - Fuente: Elaboración propia

**Configuración**

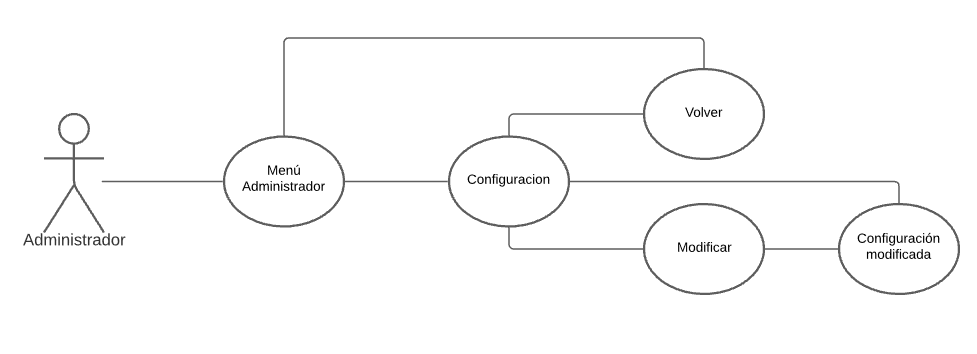
****

Imagen 20 - Título: Menú Configuración - Fuente: Elaboración propia

**DIAGRAMA DEL MODELO DE DATOS**

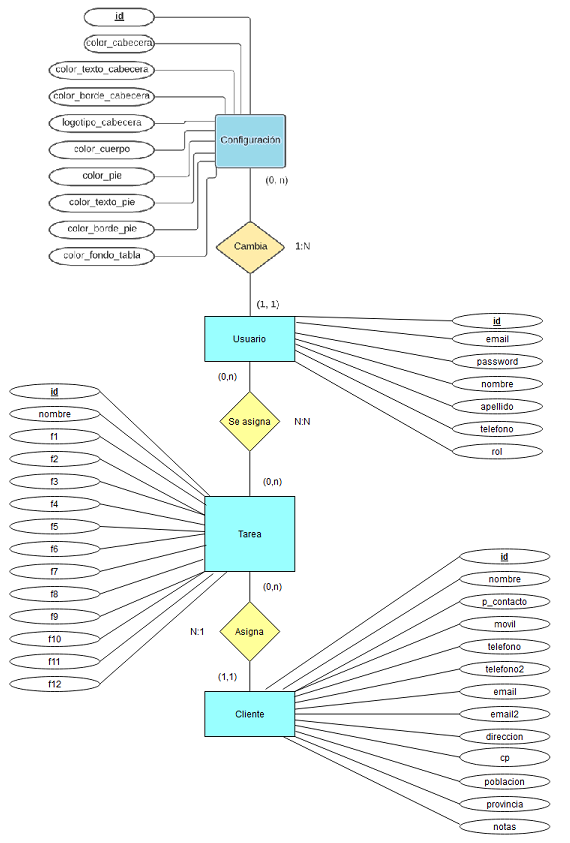
****

Imagen 2 - Título: Diagrama del modelo de datos - Fuente: Elaboración propia

**9.- DISEÑO**

**DISEÑO INICIAL**

Diseño inicial de las pantallas:

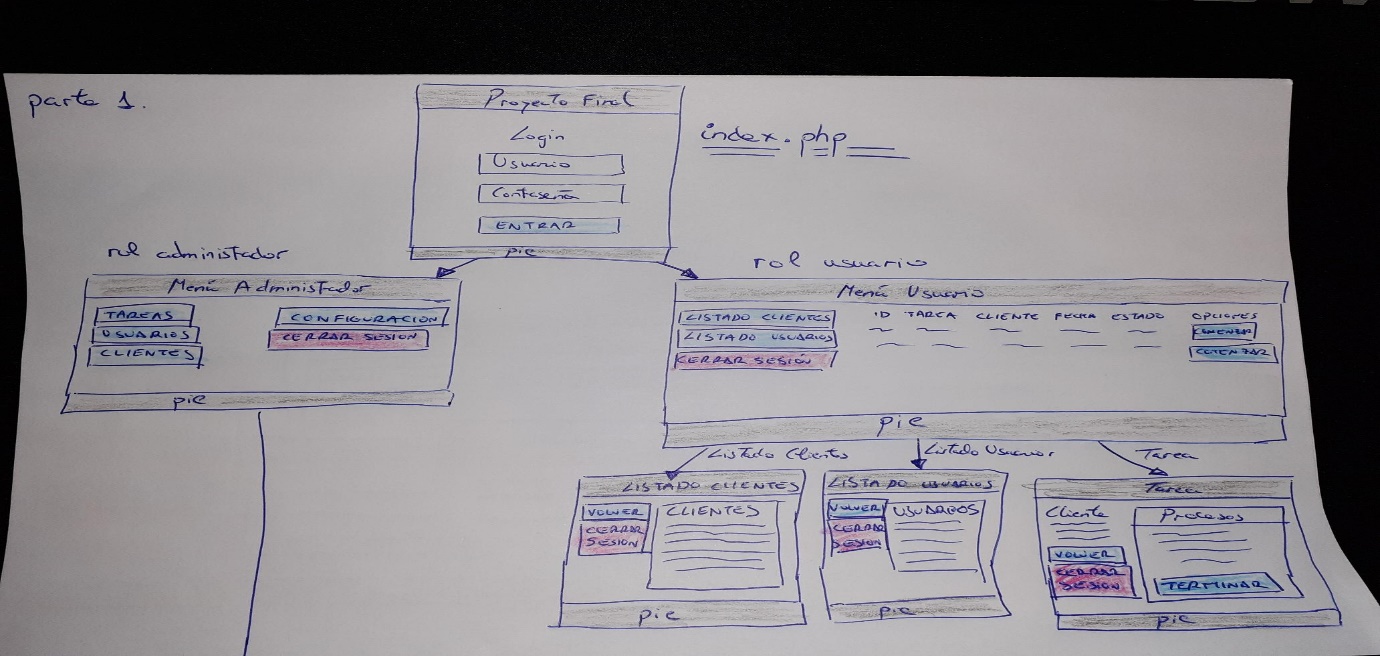
****

Imagen 22 - Título: Boceto parte 1 - Fuente: Elaboración propia

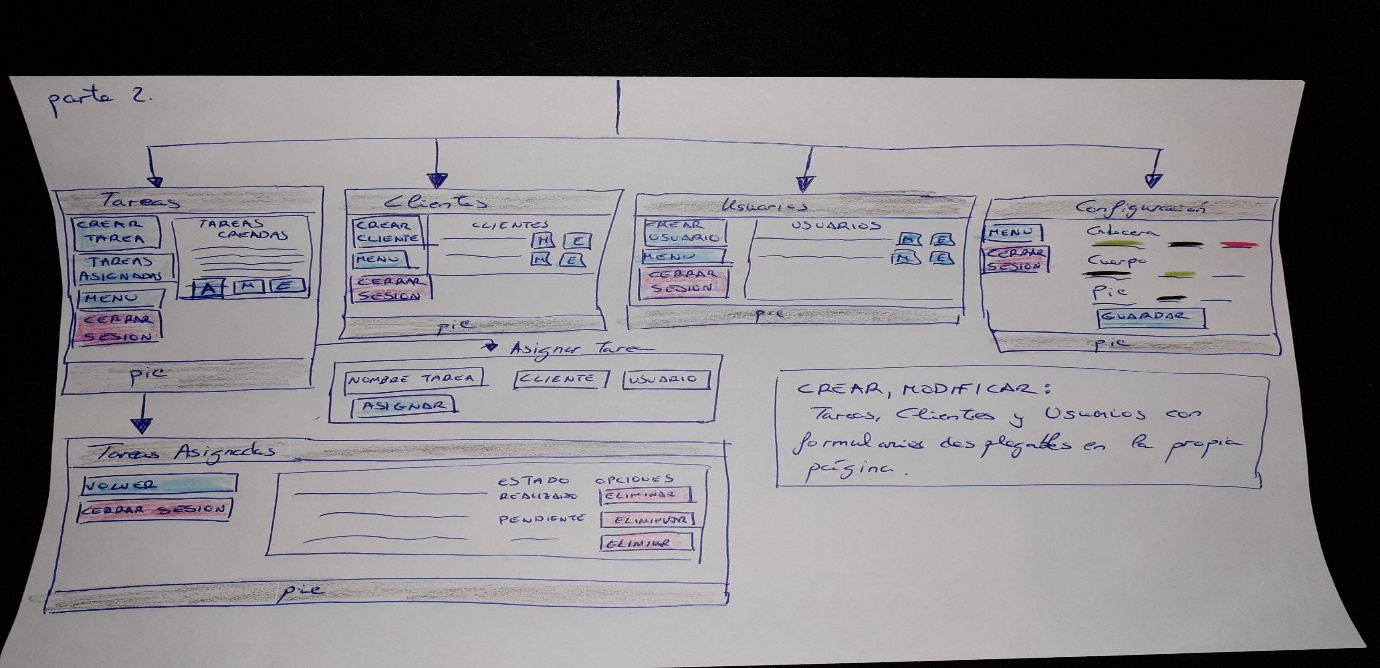
****

Imagen 23 - Título: Boceto parte 2 - Fuente: Elaboración propia

En la pantalla Tareas, si se pulsa en crear, modificar o asignar, se despliega un formulario en la misma pantalla.

En la pantalla Clientes, si se pulsa en crear o modificar, se despliega un formulario en la misma pantalla.

En la pantalla Usuarios, si se pulsa en crear o modificar, se despliega un formulario en la misma pantalla.

En algunas opciones, como TAREAS ASIGNADAS, LISTADO DE CLIENTES y LISTADO DE USUARIO, se dirige a otra página por el hecho de no sobrecargar de información la página y de esta manera que sea el visionado más claro.

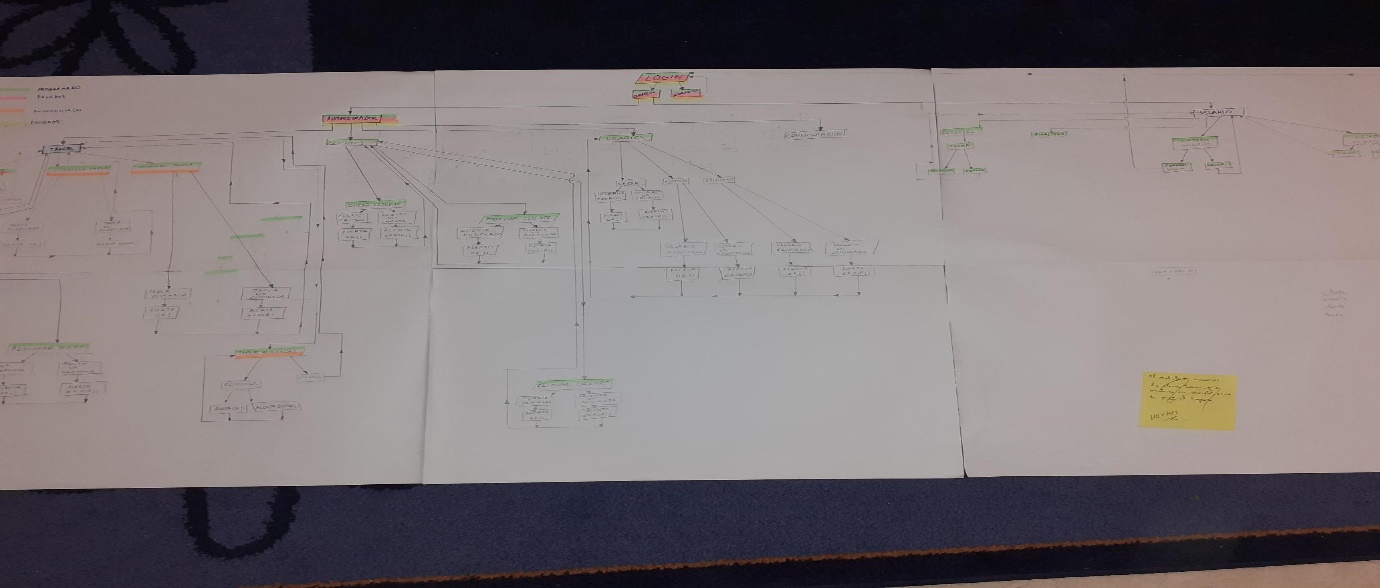
****

Imagen 24 - Título: Boceto 3 - Fuente: Elaboración propia

**EXPLICACION DETALLADA**

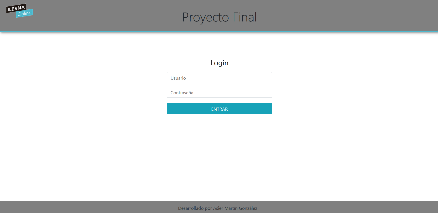
Puesto que el objetivo principal de la aplicación, es la asignación de tareas por parte del administrador hacia los usuarios, la aplicación cuenta con una serie de menús que hacen fácil el desempeño de su función.

Imagen 25 - Título: Formulario Login - Fuente: Elaboración propia

Comenzando por el menú del Login, el cual es necesario para que nadie pueda acceder a la aplicación si no tiene permiso. Este formulario de Login recogerá el email y la contraseña del usuario y los cotejará con los datos guardados en la tabla “Usuarios”. En caso de que los datos no coincidan con los guardados, será redirigido de nuevo al formulario de Login. Si los datos cotejados son correctos, se creará una sesión que podrá ser “administrador” o “usuario” dependiendo del rol que tenga guardado en la tabla “usuarios”. Si el rol es “administrador”, se redirigirá al “Menú Administrador”. Si el rol es “usuario”, se redirigirá al “Menú Usuario”.

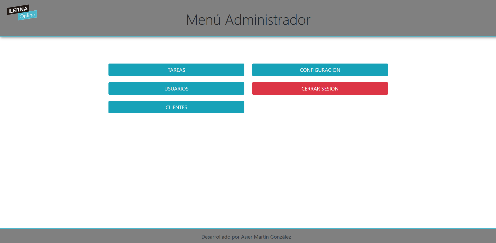
Vamos a centrarnos ahora en el caso del administrador. El “Menú Administrador” contará con 5 botones (en un futuro se podrían incluir más en función de las necesidades). Serán de diseño actual y lectura fácil. Se busca la eficacia y efectividad en la aplicación por encima del diseño, ya que no es una aplicación cara al público, si no para uso interno.

Imagen 26 - Título: Menú Administrador - Fuente: Elaboración propia

El botón **TAREAS** enviará al administrador a otro menú denominado “Tareas”. Desde este menú, el administrador podrá asignar, modificar, eliminar y crear tareas.

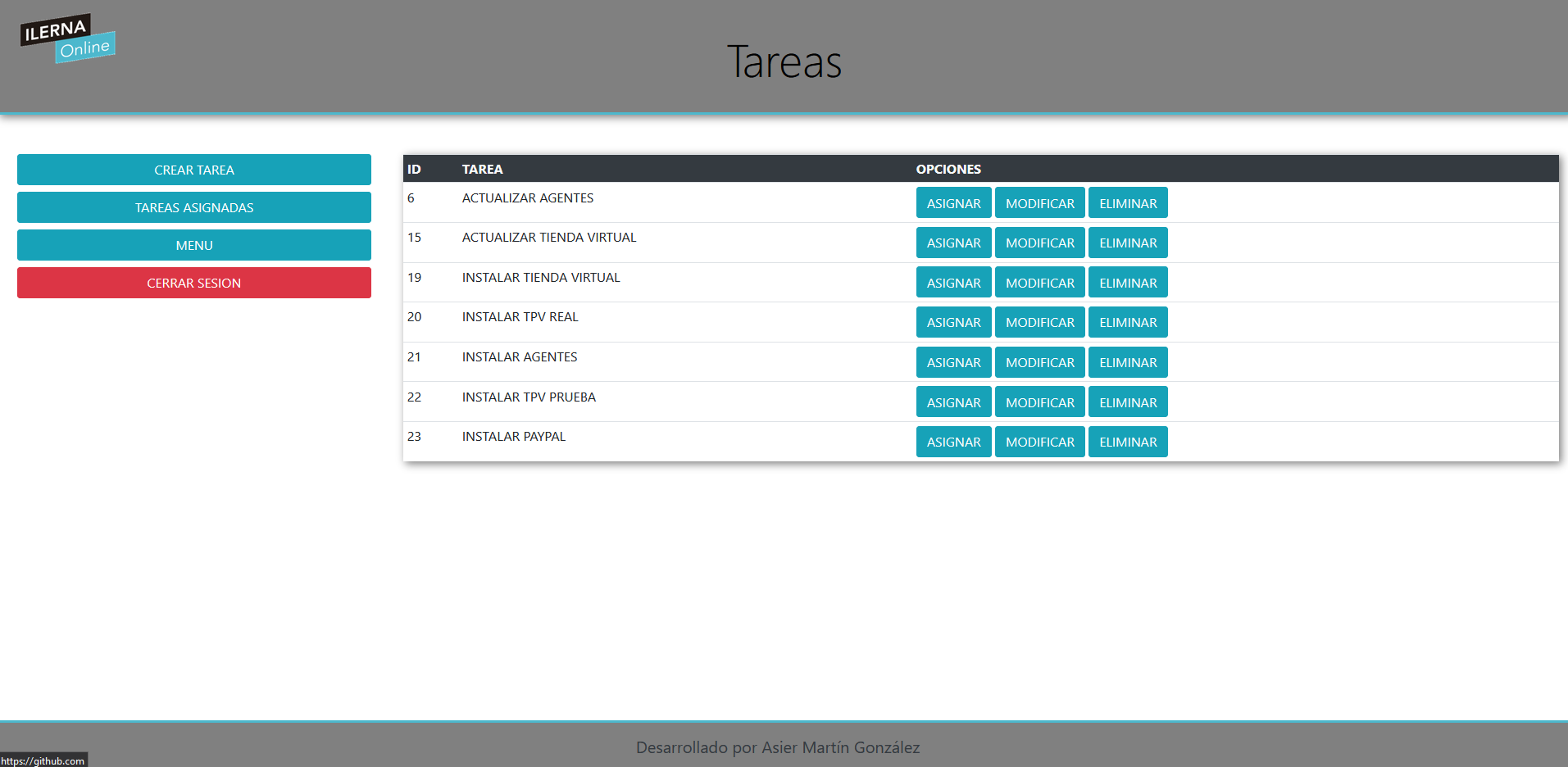


Imagen 27 - Título: Menú Tareas - Fuente: Elaboración propia

También podrá asignar las tareas a un usuario. Contará además con un botón para volver al “Menú Administrador” y otro botón para cerrar sesión. Si existen tareas creadas, se mostrarán en una tabla y a la derecha de cada tarea aparecerán los botones para asignarla, modificarla o eliminarla. A la izquierda de la misma, aparecerán los botones para crear una tarea, ver las tareas asignadas, volver al “Menú Administrador” y cerrar sesión. Si se pulsa sobre el botón ASIGNAR se desplegará un formulario que contendrá el nombre de la tarea y se podrá seleccionar un cliente sobre el cual realizar el trabajo y un usuario al cual adjudicar dicha tarea. Este formulario se replegará al pulsar sobre ASIGNAR o sobre el símbolo (x). Si se pulsa sobre ASIGNAR nos mostrará una alerta indicando si se ha podido asignar la tarea a un usuario o si ha ocurrido un error y no se ha podido asignar la tarea.

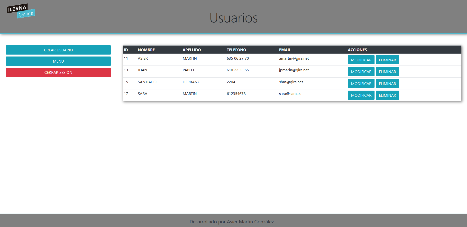
Si se pulsa sobre el botón MODIFICAR se desplegará un formulario con los datos de la tarea seleccionada. El formulario contará con los campos ID, Nombre tarea y 12 apartados más en donde se podrán poner los pasos a llevar a cabo para realizar la tarea. Este formulario se replegará al pulsar sobre el botón MODIFICAR o al pulsar sobre el símbolo (x). Si se modifica la tarea nos mostrará una alerta indicando si se ha podido modificar o si ha ocurrido un error.Si se pulsa sobre el botón ELIMINAR, la aplicación nos lanzará una alerta indicando si se ha podido eliminar la tarea seleccionada o si ha ocurrido un error al intentar eliminar la tarea. Si se pulsa sobre el botón CREAR TAREA se despliega un formulario en el que incluir el nombre de la tarea y hasta 12 pasos o funciones que el usuario deberá de seguir. Una vez se pulse sobre el botón CREAR la aplicación lanzará una alerta indicando si se ha podido crear la tarea o si no se ha podido crear la tarea.

Imagen 28 - Título: Tareas asignadas - Fuente: Elaboración propia

Si se pulsa sobre el botón TAREAS ASIGNADAS se envía al administrador a otra página (para descargar de contenido em menú Tareas).

En esta página con título Tareas Asignadas se muestran en una tabla las tareas asignadas a los usuarios y se visualiza el estado de la misma. Este estado puede ser pendiente o realizado. A la derecha del estado se dispone de un botón para eliminar la tarea asignada se si desea. En caso de pulsarlo, la aplicación lanzará una alerta indicando si se ha podido eliminar la tarea o si no se ha podido eliminar la tarea.

En esta página, también se dispone de un botón VOLVER que dirigirá al menú Tareas y otro botón CERRAR SESION para que el administrador pueda cerrar la sesión.

Volviendo al Menú Administrador, se dispone de otro botón USUARIOS el cual dirigirá al menú Usuarios.

Nuevamente se dispone de los botones CREAR USUARIO, MODIFICAR, ELIMINAR y CERRAR SESION además de una tabla donde se observan los usuarios ya creados.

CREAR USUARIO despliega un formulario que solicita varios parámetros, todos requeridos, para poder crear un nuevo usuario. En caso de pulsar sobre el botón CREAR la aplicación envía de nuevo al administrador al menú Usuarios y lanzará una alerta indicando si se ha podido crear el usuario o no se ha podido crear el usuario.

MODIFICAR despliega un formulario con los datos del usuario seleccionado, desde el cual se pueden modificar los datos. Una vez se pulse sobre el botón MODIFICAR, la aplicación lanzará una alerta indicando si se ha podido modificar el usuario o si no se ha podido modificar el usuario.

ELIMINAR elimina el usuario seleccionado y lanza una alerta indicando si se ha podido eliminar el usuario o si no se ha podido eliminar el usuario.

El botón MENU envía al administrador al Menú Administrador y el botón CERRAR SESION, cierra la sesión del administrador y lo envía al formulario de Login.

De nuevo en el Menú Administrador, el botón CLIENTES envía al administrador a la página con el título de Clientes. Una vez más, se encuentran los botones CREAR CLIENTE, MODIFICAR, ELIMINAR, MENU y CERRAR SESION.

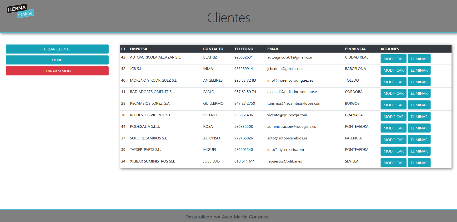


Imagen: 29 - Título: Clientes - Fuente: Elaboración propia

CREAR CLIENTE despliega un formulario con el título Nuevo Cliente en el cual los siguientes campos son obligatorios: Empresa, Persona de contacto, Teléfono 1, Email 1, Dirección, CP, Población y Provincia. El resto de los campos no son requeridos ya que se puede dar el caso, por ejemplo, de que el cliente no disponga de más de un número de teléfono o de más de un email. Una vez se pulse en CREAR la aplicación lanzará una alerta indicando si se ha podido crear el nuevo cliente o si no se ha podido crear el cliente. Si se ha creado el cliente, esta aparecerá de manera automática en la tabla. MODIFICAR despliega un formulario con los datos del cliente seleccionado. Desde el formulario se podrá modificar los datos del cliente y guardarlos pulsando sobre el botón MODIFICAR. La aplicación lanzará una alerta indicando si se ha podido modificar el cliente o si no se ha podido modificar el cliente. En caso afirmativo, los cambios se verán reflejados en la tabla.

El botón ELIMINAR elimina el cliente seleccionado y lanza una alerta indicando si se ha podido eliminar el cliente o si no se ha podido eliminar el cliente. En caso afirmativo, el cliente desaparecerá de la tabla. El botón MENU envía al administrador al Menú Administrador y el botón CERRAR SESION cierra la sesión del administrador y lo envía al formulario de Login.

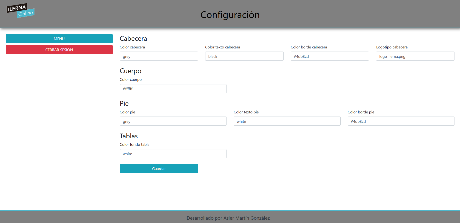
Nuevamente en el Menú Administrador, el botón CONFIGURACION envía al administrador a la página con título Configuración.

Imagen 30 -Título: Configuración - Fuente: Elaboración propia

Esta página cuenta con un formulario para poder cambiar algunos parámetros del diseño. A día de presentar el proyecto, los cambios que haga el administrador serán aplicables para todos los usuarios. Los parámetros que se pueden modificar son: color fondo cabecera, color texto cabecera, color borde cabecera, logotipo cabecera, color cuerpo, color fondo pie, color texto pie, color borde pie y color fondo tabla.

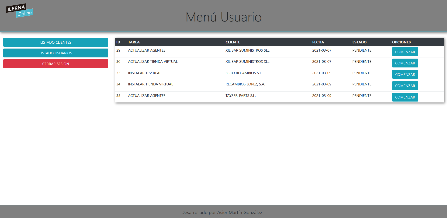
Para el futuro se baraja la opción de que cada usuario pueda modificar los parámetros a su gusto o que se pueda contar con 2 temas: uno claro y otro oscuro. El botón MENU envía al administrador al Menú Administrador y CERRAR SESION cierra la sesión del administrador y lo envía al formulario de Login. De nuevo en el Menú Administrador, y ya, por último, se encuentra el botón CERRAR SESION. Este botón cierra la sesión del administrador y lo envía al formulario de Login.

Imagen 31 – Título: Menú Usuario – Fuente: Elaboración propia

Supuesto del caso en el que al logearse el empleado tenga rol de usuario:

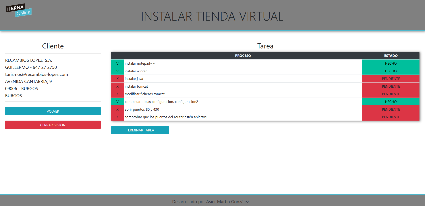
Se creará una sesión de usuario y se le enviará a la página con el título Menú Usuario. El Menú Usuario consta de una tabla en la que se muestra las tareas asignadas a ese usuario. Cada tarea cuenta con un botón COMENZAR el cual envía al usuario a una página con el título del nombre de la tarea seleccionada. En esta página, en el lado izquierdo, aparecen los datos del cliente sobre el que se va a realizar el trabajo. A la derecha se despliega una lista con los pasos a cumplir de la tarea. Dispone de la opción de ir chekando los pasos que se van realizando. Esto es opcional, aunque de cara a un futuro, se baraja la opción de que haya que marcar todos los campos para poder dar por finalizada la tarea.

Imagen 32 -Título: Tarea asignada - Fuente: Elaboración propia

Si el usuario pulsa sobre el botón TERMINAR TAREA al administrador le aparecerá como tarea finalizada y al usuario se le borrará automáticamente en el Menú Usuario.De vuelta al Menú Usuario. A parte de la tabla a la derecha en la que aparecen las tareas que el usuario tiene pendiente, en el lado izquierdo aparecen los botones LISTADO CLIENTES, LISTADO USUARIOS y CERRAR SESION.

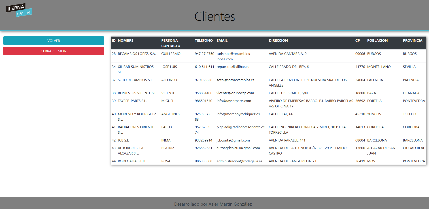
LISTADO CLIENTES envía al usuario a una página con título Clientes, donde podrá apreciar de forma clara, un listado con todos los clientes de la empresa. En este caso, el usuario no tiene acceso a crear, modificar o eliminar clientes, ya que

Imagen 33 - Título: Listado de clientes - Fuente: Elaboración propia

estas acciones solo pueden ser llevadas a cabo por el administrador. Los clientes se muestran ordenados alfabéticamente. Esta página cuenta con un botón VOLVER, el cual dirige al usuario al Menú Usuario y el botón CERRAR SESION, el cual cierra la sesión del usuario y le dirige al formulario de Login..

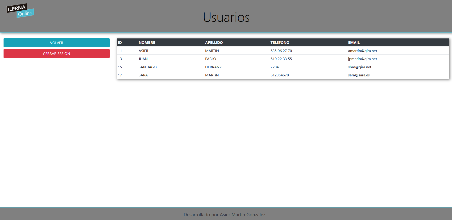
En el Menú Usuario también se dispone de un botón LISTADO USUARIOS. Si se pulsa el botón, el usuario es dirigido a la página con el título Usuarios. En esta página se despliega una tabla con los datos del resto de los usuarios, muy útil para obtener el teléfono o el email de uno de ellos.

Imagen 34 -Título: Listado de usuarios - Fuente: Elaboración propia

Los usuarios se muestran ordenados alfabéticamente por el nombre. Esta página dispone de un botón VOLVER, el cual manda al usuario al Menú Usuario, y un botón CERRAR SESION, que cierra la sesión del usuario y lo envía al formulario de Login.

Básicamente este es el funcionamiento de la aplicación t\_GIRA. De cara al futuro se podrían añadir más funcionalidades, como un chat interno y la opción de que se genere un email automático que informe al cliente que el trabajo ha sido realizado.

Toda la aplicación se ha diseñado pensando en que se adapte principalmente a tres tamaños de pantalla. Ordenadores, tables y móviles. Bien es cierto, que en este apartado no se ha profundizado mucho, ya que la aplicación está pensada para usarse desde los ordenadores y el hecho de hacerla adaptable ha sido por un requisito del proyecto más que por un tema funcional.

**CLASES**

De cara a realizar una aplicación web con programación orientada a objetos, se han creado cuatros clases para optimizar la programación y poder controlar mejor las funcionalidades de cada una de las partes de la aplicación.

**Usuario**

Con esta clase se controla la creación, modificación y eliminación de usuarios. La clase está creada en el archivo Usuario.php

Cuenta con un método constructor, los getters y setters necesarios para utilizar los atributos de clase, y con las funciones crearUsuario(), modificarUsuario() y eliminarUsuario(). Para separar el código y hacerlo más legible, esta clase es llamada desde el archivo opciones\_usuarios.php

**Tarea**

La clase Tarea también es capaz de crear, modificar y eliminar tareas. Esta clase se encuentra en el archivo Tarea.php

El método constructor recoge catorce atributos de clase y está pensada la idea que en un futuro el administrador pueda elegir la cantidad de atributos que quiere que tenga el constructor. La clase cuenta con su método constructor, las funciones getters y setters, y las funciones crearTarea(), modificarTarea() y eliminarTarea()

Para separar el código y hacerlo más comprensible, se llama a la clase Tarea desde el archivo opciones\_tareas.php Es en este archivo, donde se llama a las diferentes funciones de la clase Tarea según las necesidades.

Se podría haber incluido el código de opciones\_tareas.php en Tarea.php, pero el archivo era muy extenso y complicaba su mantenimiento.

**Cliente**

Esta clase está albergada en el archivo Cliente.php Con ella se pueden crear, modificar y eliminar clientes.

Dispone de un método constructor, los métodos getters y setters necesarios y las funciones crearCliente(), modificarCliente() y eliminarCliente().

Al igual que en las otras clases, con el objetivo de hacer el código más legible y ordenado, la clase es llamada desde el archivo opciones\_clientes.php

Desde este archivo son demandadas las funciones de la clase Cliente en función de las necesidades de la aplicación.

**Configuracion**

Por último, nos encontramos con la clase Configuracion. Esta clase es la encargada de modificar la configuración gráfica de la aplicación. Para este proyecto se ha otorgado al administrador la potestad de poder variar distintos parámetros, pero más adelante, se baraja la posibilidad de dejar solamente un modo claro y un modo oscuro.

La clase Configuracion, cuenta con su método constructor, los métodos getters y setters necesarios, y con la función modificarConfiguracion().

Al igual que en las clases anteriores, se ha optado por llamar a la clase desde otro archivo. En esta ocasión el archivo es opciones\_configuracion.php

**FUNCIONES**

Ya se han descrito las funciones que porta cada una de las clases. A parte de las funciones anteriores, también se han creado diversas funciones fuera de las clases.

Dentro del archivo alertas.php se encuentran dos funciones. Estas funciones son: eliminar\_alertas() y mostrar\_alertas().

La función eliminar\_alertas() elimina las sesiones necesarias para que no se muestren las alertas.

La función mostrar\_alertas() se utiliza en los archivos en los que es necesario que se vayan mostrando las distintas alertas para indicar a los usuarios que las opciones se han llevado a cabo o si ha sucedido un error.

**FICHEROS**

Para organizar los distintos ficheros y archivos, se ha optado en separar los archivos en función de la tarea general que realizan.

Los archivos que son pantallas de la aplicación se han incluido en el directorio raíz de la aplicación. Los archivos que son reutilizados constantemente, se han incluido en la carpeta includes. Dentro de la carpeta imagenes, se han incluido todas las imágenes. En la carpeta clase, se encuentran las clases Cliente, Configuracion, Tarea y Usuario. Y por último, dentro de la carpeta bbdd, se encuentran todos los archivos que interactúan con la base de datos.

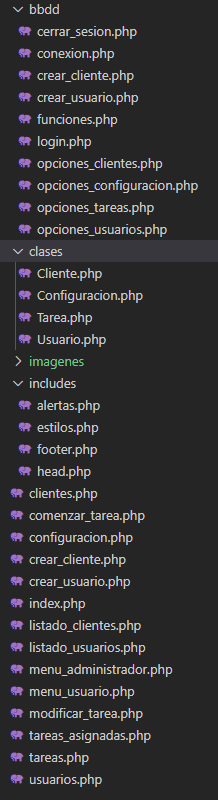
****

Imagen 35 - Título: Ficheros aplicación - Fuente: Elaboración propia

**10.- DESPLIEGUE Y PRUEBAS**

**DESPLIEGUE**

**PRUEBAS**

Las pruebas han sido realizadas siguiendo el proceso de “Pruebas de caja negra”. Se han realizado en todas las pantallas de la aplicación web.

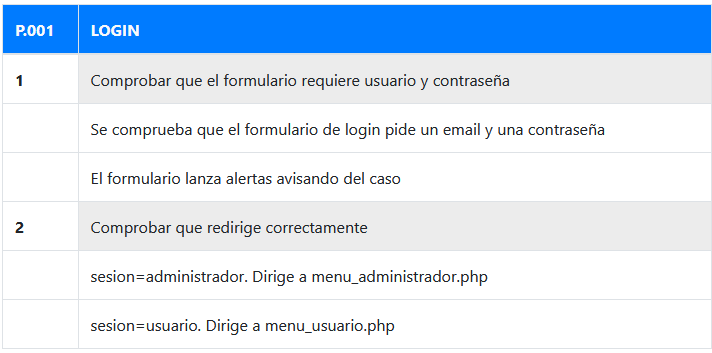
****

Imagen 36 - Titulo: p.001 - Fuente: Elaboración propia



Imagen 37 - Título p.002 - Fuente: Elaboración propia

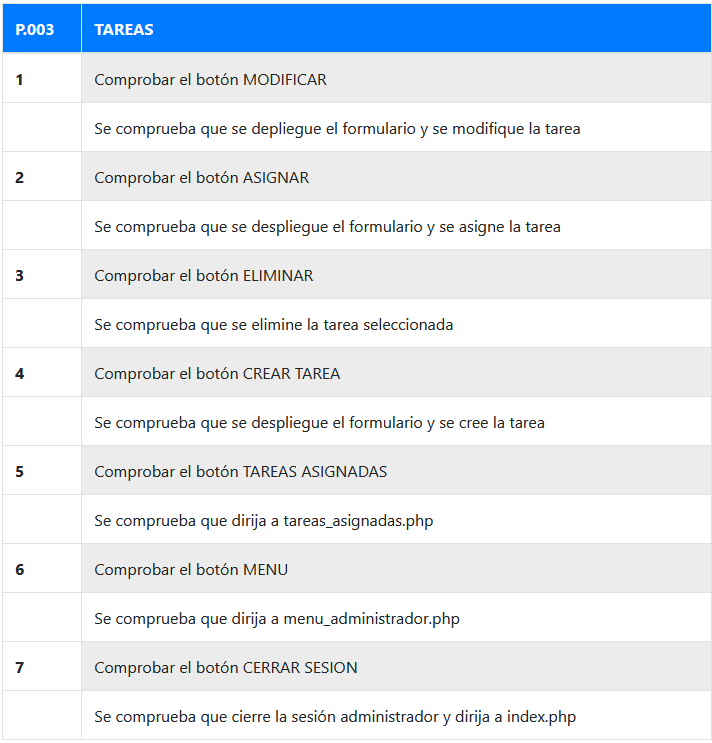


Imagen 38 - Título: p.003 - Fuente: elaboración propia

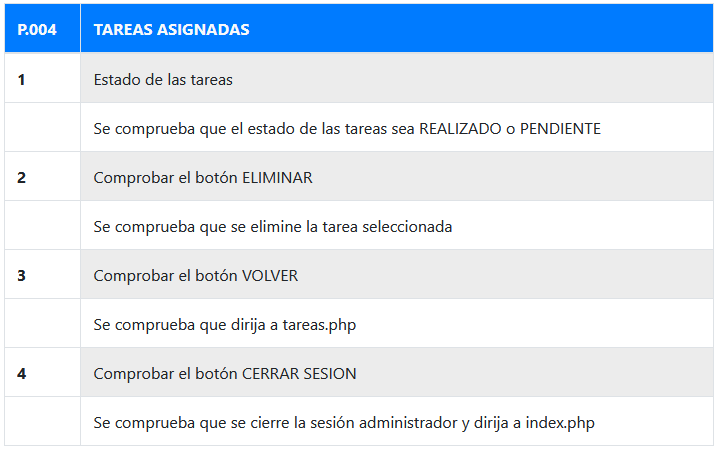


Imagen 39 - Título: 0.004 - Fuente: Elaboración propia

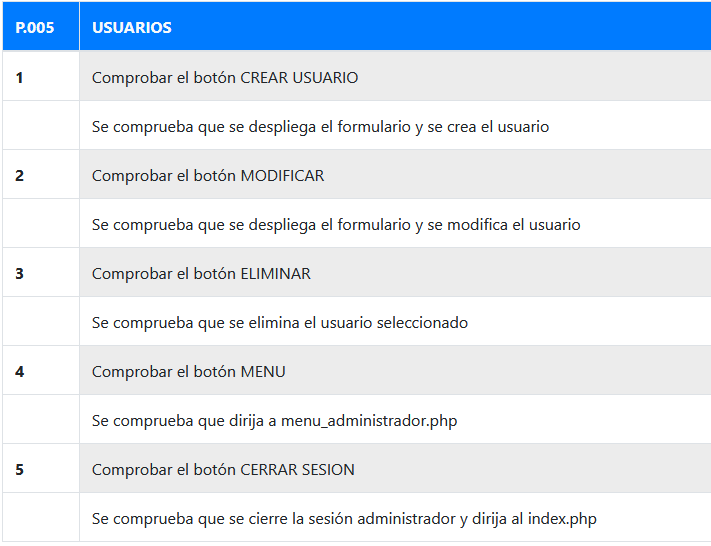


Imagen 40 - Título: p.005 - Fuente: Elaboración propia

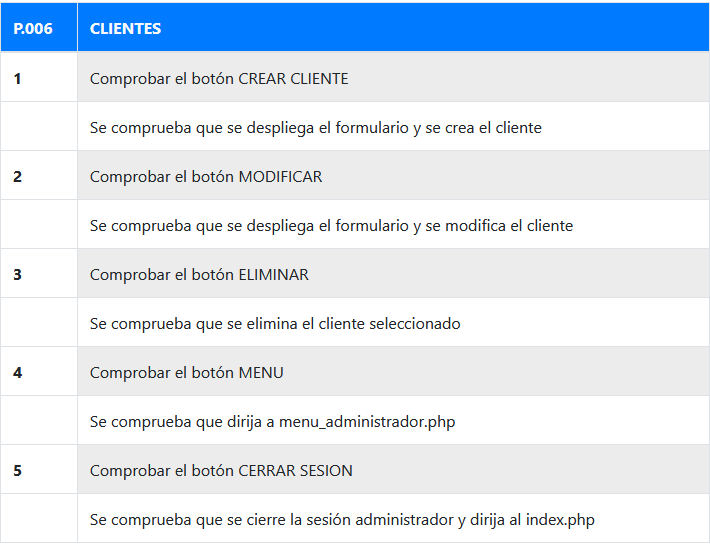


Imagen 41 - Título: p.006 - Fuente: Elaboración propia

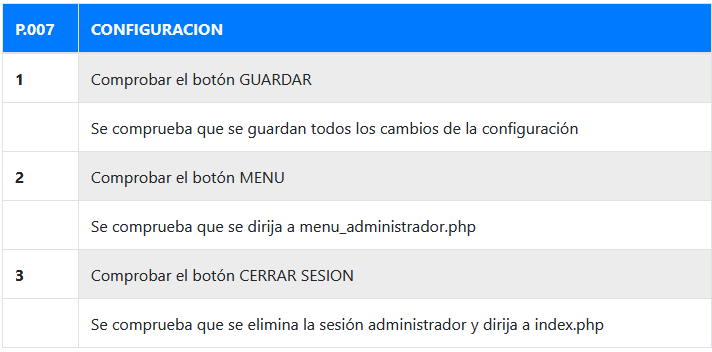


Imagen 42 - Título: p.007 - Fuente: Elaboración propia

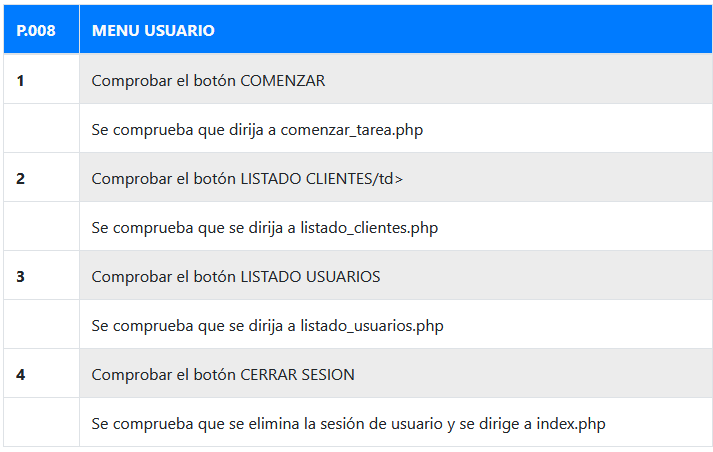


Imagen 43 - Título: p.008 - Fuente: Elaboración propia

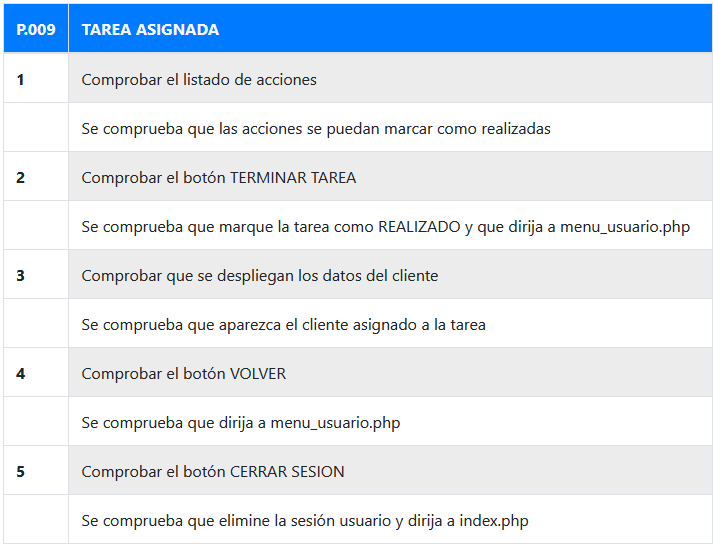


Imagen 44 - Título: p.009 - Fuente: Elaboración propia

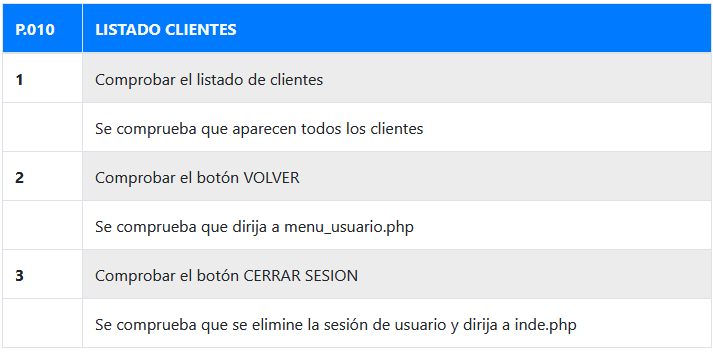


Imagen 45 - Título: p.010 - Fuente: Elaboración propia

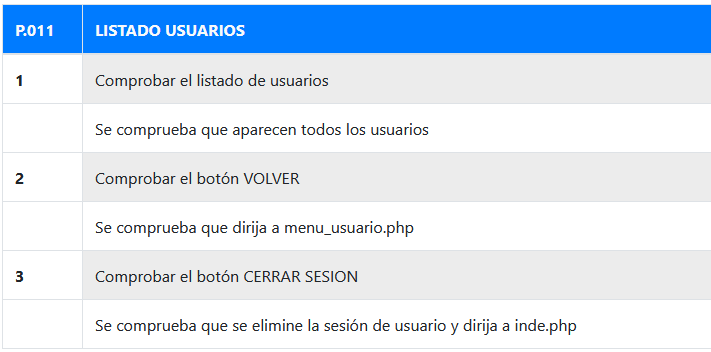


Imagen 46 - Título: p.011 - Fuente: Elaboración propia

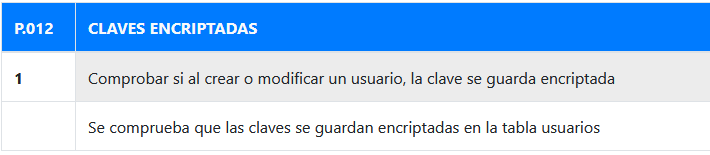


Imagen 47 - Título: p.012 - Fuente: Elaboración propia

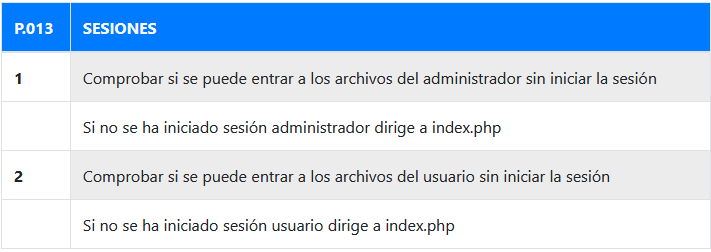


Imagen 48 - Título: p.013 - Fuente: Elaboración propia

**11.- CONCLUSIONES**

**OBJETIVOS ALCANZADOS**

**CONCLUSIONES**

**12.- VIAS FUTURAS**

Según se ha ido realizando el proyecto y probando, se ha visto que hay bastantes posibilidades de ir mejorando la aplicación en un futuro. Estos cambios pueden ser tanto de las funcionalidades ya programadas, como la creación de nuevos módulos para ampliar la funcionalidad de la aplicación.

**- Formulario de Login:** Permitir que se pueda guardan las sesiones por un tiempo determinado, para evitar que los usuarios tengan que volver a logearse.

**- Buscador clientes:** El listado de clientes se puede mejorar incluyendo un buscador para que cuando haya un número elevado de clientes, sea más sencillo encontrar a un cliente determinado.

- **Paginación:** En los listados de clientes, usuarios y tareas, se podrá incluir una paginación para evitar el uso excesivo del scroll.

**- Tareas para el administrador:** Actualmente, el administrador solo puede asignar tareas a los usuarios. Se pretende que el administrador pueda adjudicarse tareas también a si mismo.

**- Preguntar antes de llevar una acción a cabo:** Para evitar ejecutar una acción no deseada, se incluirá una pregunta de refuerzo, para evitar llevar a cabo una acción accidentalmente.

**- Crear una tabla para guardar todas las tareas realizadas:** Actualmente, una vez que el administrador elimina una tarea, esta desaparece del sistema. Se pretende crear una nueva tabla, donde se guarden todas las tareas que se han eliminado, por si en el futuro hay que realizar alguna consulta.

**- Configuración:** Se baraja la posibilidad de eliminar el sistema de configuración actual, y dejar a opción del usuario la posibilidad de poder elegir entre un modo claro o un modo oscuro.

**- Funciones eliminar\_alertas() y mostrar\_alertas():** Se pretende pasar parámetros a las funciones para reducir las líneas de código de cada una de ellas y hacer un código más efectivo y optimizado.

También se quiere incluir un diseño más moderno y dinámico.

**- MVC:** Separar los archivos utilizando el sistema Módelo-Vista-Controlador.

**- Cotejamiento campos formularios y tamaño campos de las tablas:** Se añadirá max-length a los formularios para controlar que el tamaño de los campos introducidos no supere al tamaño de lo otorgado a su campo en la tabla correspondiente.

**13.- GLOSARIO**

**CRUD**

Framework

HTML

CSS

JQuery

JavaScript

Responsive

Frontend

Backend

Login

Logearse

MySQL

PHPMyAdmin

UML

GitHub

SO (Sistema Operativo)

Lucidchart

Visual Studio Code

Scroll

Lucidchart

CRUD

MVC

max-length

**14.- BIBLIOGRAFIA / WEBGRAFIA**

-Udemy: Desarrollo de sistemas web en PHP, POO y MySQL

[www.udemy.com/course/desarrollo-sistemas-web-php-poo-mysql-jquery-ajax-bootstrap](http://www.udemy.com/course/desarrollo-sistemas-web-php-poo-mysql-jquery-ajax-bootstrap)

-Píldoras informáticas: Curso de JavaScript

[www.pildorasinformaticas.es/course/javascript-desde-0/](http://www.pildorasinformaticas.es/course/javascript-desde-0/)

-w3schools.com (Bootstrap4)

[www.w3schools.com/bootstrap4/default.asp](http://www.w3schools.com/bootstrap4/default.asp)

-w3schools.com (JQuery)

[www.w3schools.com/jquery/default.asp](http://www.w3schools.com/jquery/default.asp)

-w3schools.com (JavaScript)

[www.w3schools.com/js/default.asp](http://www.w3schools.com/js/default.asp)

-SQL. Los fundamentos del lenguaje. 2ª edición. Editorial ENI.

-DiagramasUML

<https://diagramasuml.com/diagrama-de-clases/>

**15.- ANEXOS**

RESPONSIVE

DIAGRAMAS

