

# MEMORIA FINAL DE PROYECTO

## Tasker

Ciclo Formativo Grado Superior  
Desarrollo de Aplicaciones Web Bilingüe

Autor

David Tobalina Rodríguez

TUTOR

Sergio Báez Ferrer

COORDINADOR

Xavier Ganzábal García

CURSO

2020/ 2021

I.E.S CLARA DEL REY



## Índice

1. Título
2. Introducción
3. Alcance del proyecto y análisis previo
4. Requisitos y objetivos
  - 4.1. Requisitos
  - 4.2. Objetivos
5. Viabilidad y soluciones escogidas
  - 5.1. La web
  - 5.2. Estudio preliminar SEO
  - 5.3. Planificación de tareas
6. Diseño
  - 6.1. Maqueta
  - 6.2. Estética del sitio
  - 6.3. Mapa del sitio
  - 6.4. Modelo de entidad-relación
  - 6.5. Diagrama de casos de uso
  - 6.6. Modificaciones y aprobación del diseño
7. Herramientas
  - 7.1. Servidor XAMPP
  - 7.2. PHP
  - 7.3. JavaScript
  - 7.4. AJAX
  - 7.5. SQL
  - 7.6. HTML
  - 7.7. CSS
  - 7.8. Visual Studio Code
  - 7.9. Pencil
  - 7.10. PHP Mailer
  - 7.11. Repositorio GitHub
8. Base de datos
  - 8.1. Diagrama
  - 8.2. Almacenamiento
  - 8.3. Datos de prueba
9. Estructura y exposición del código
  - 9.1. Estructura de archivos
  - 9.2. Inicio
  - 9.3. Aplicación
10. Manual de usuario
11. Conclusión
12. Referencias

## 1. Título

El título de este proyecto es “Tasker”.

## 2. Introducción

A continuación, se va a desarrollar la memoria del proyecto de fin de Grado Superior de Desarrollo de Aplicaciones Web (2020-2021). Dicha memoria consiste en la documentación y planificación correspondiente al trabajo realizado en el proyecto que ha sido elaborado durante el periodo comprendido entre los meses de abril y mayo de 2021.

En este documento encontraremos diversas secciones donde se detalla desde la fase de planificación hasta la fase de implementación y puesta en marcha.

En esta memoria se explica la realización de un sitio web en base a lo aprendido durante el curso y plasmarlo de manera práctica, es decir, sintetizar lo aprendido hasta la fecha para convertir una idea en un producto tangible.

Al tratarse de un grupo de modalidad bilingüe se ha decidido desarrollar la página web en inglés. Aun así, la documentación correspondiente está en idioma español, para facilitar la lectura.

He decidido crear un sitio web que ayude a los usuarios a organizarse en su día a día de manera que puedan emplear su tiempo de manera más productiva y eficiente. Esto se consigue poniendo a su disposición una variedad de herramientas que se detallan más adelante.

En los siguientes apartados especificaré los diferentes aspectos, así como los objetivos del trabajo que se pretenden en la creación de este sitio web.

### Introduction

Next, the memory of the final project of the Higher Degree in Web Application Development (2020-2021) will be developed. Said report consists of the documentation and planning corresponding to the work carried out in the project that has been prepared during the period between the months of April and May 2021.

The project focuses its effort on the creation of a product starting from the pre-analysis and planning to the implementation and start-up stage.

This report explains how to create a website based on what has been learned throughout the training cycle completed during the course and translate it in a practical way, that is, synthesize what has been learned to date to turn an idea into a tangible product.

As it is a bilingual group, it has been decided to develop the website in English. Even so, the corresponding documentation is in Spanish, to facilitate reading.

I have decided to create a website that helps users organize their day to day so that they can use their time more productively and efficiently. This is accomplished by making a variety of tools available to you, which are detailed below.

In the following sections I will specify the different aspects, as well as the objectives of the work that are intended in the creation of this website.

### 3. Alcance del proyecto y análisis previo



Figura 3.: Fuente: <https://okdiario.com/img/2019/01/24/como-hacer-un-estudio-de-viabilidad.jpg>

Este proyecto pretende crear una aplicación web de organizador de tareas, así como agenda de eventos tanto para su consulta a través de un ordenador personal como de otros dispositivos ya sean tabletas o teléfonos móviles.

El objetivo principal del proyecto consiste en crear dicha página web que permitirá una herramienta accesible, fácil de interpretar, rápida, cómoda y eficaz de consulta de tareas y organizador de agenda, consistiendo en un sistema web sencillo y que pueda estar en constante crecimiento.

Esta página web ha de ser sencilla de utilizar en todos los dispositivos utilizados habitualmente. Para conseguir una buena navegación a través de la página web se planificará un diseño cómodo, agradable y fácil de comprender por todos los usuarios que se conecten mediante un dispositivo móvil u ordenador a dicha aplicación.

Una vez explicado el contexto del proyecto pasaremos a detallar la importancia de una web de tareas, razón que me ha llevado a desarrollar esta web.

Las agendas o planificadores correctamente empleados ofrecen una metodología de trabajo más organizada y analítica. Son una herramienta de vital importancia para aumentar la productividad y mejor aprovechamiento del tiempo. Ya sea una agenda de carácter personal, de trabajo o profesional, están planteadas para organizar y guiar a sus usuarios a la hora de alcanzar sus objetivos, pero para ello deben ser utilizadas de forma correcta y dichas agendas deben ser funcionales, sino no cumplirán su propósito.

Es muy habitual que las personas inicien tareas que no finalizan, o bien elaboran planes u objetivos que finalmente no llevan a cabo bien por olvidos, o por un exceso de tareas diarias que no se saben priorizar. En estos casos, un organizador profesional o personal ayuda a enumerar y priorizar propósitos siendo una aplicación que podamos consultar en nuestro ordenador personal o móvil una ventaja añadida. Aquellas personas que presentan además estrés laboral (sobre todo personas autónomas) al contar con un alto grado de responsabilidad y un cúmulo sinfín de tareas, un planificador les va a permitir una organización racional y eficaz del tiempo.

Las agendas también permiten además la recopilación de datos e informaciones con las que no se necesita saturar el cerebro, tales como horarios de reuniones o entrevistas, eventos, etc. Contar con un espacio para ello en nuestro propio ordenador o dispositivo móvil será de gran efectividad.

La gestión del tiempo en la actualidad es imprescindible, crece cada vez más su interés en la actualidad por la vida tan activa y caótica que tenemos. Cada vez abarcamos más obligaciones y complicaciones en nuestra vida diaria que nos obligan a gestionar mejor nuestro tiempo.

Aunque la realidad es que el tiempo no podemos gestionarlo tal y como hacemos con nuestro dinero, ya que el tiempo fluye sin detenerse; si podemos hacer algo para intentar solucionar este dilema que es gestionar lo que hacemos nosotros con el tiempo que tenemos, es decir, gestionarnos a nosotros mismos.

Con la gestión del tiempo lo que realizamos sobre la planificación de las tareas, consiste en ir priorizando unas sobre otras. También se incluye el análisis y la definición de objetivos que deben alcanzarse en un tiempo determinado. Existen multitud de métodos de gestión del tiempo y en general, los métodos de gestión del tiempo suelen estar enfocados a ordenar las tareas por prioridad y por impacto.

También muchas veces están enfocados a evitar tareas que se pueden automatizar o a decir no a algunos compromisos que no aportan demasiado pero que consumen tiempo. Es decir, no se trata sólo de hacer lo que hay que hacer eficientemente, sino también de elegir lo que se hace con el tiempo. Una buena gestión de nuestro tiempo nos puede ayudar a que nuestra vida personal y profesional sea mejor.

Además, esta web permite la elaboración de una lista de la compra con una serie de alimentos ya incorporados, que admite añadir la cantidad que se precisa adquirir en el supermercado.

Una tarea tan cotidiana en nuestro hogar como es hacer la lista de la compra, cobra gran importancia cuando se trata por ejemplo de seguir una alimentación saludable. Si no tenemos una buena planificación y hacemos la compra de manera improvisada cuando estamos en el supermercado, propiciamos la compra de alimentos superfluos o poco saludables (ultra procesados) y nos acabamos olvidando de los alimentos saludables básicos que siempre deberían estar en nuestras casas. De manera que esta web permite adquirir hábitos de vida saludable con respecto a la alimentación con una buena organización a la hora de elaborar una lista de la compra que nos ayude a conseguir dicho objetivo

Algunas de las ventajas que se pueden obtener a la hora de planificar las compras con una lista son:

- Ahorro: al ceñirnos a lo realmente necesario y a las cantidades apuntadas, se evitará comprar alimentos superfluos
- Adherencia a una dieta saludable: con alimentos saludables se podrá conseguir una buena alimentación
- Mayor productividad: se evitarán olvidos de productos y visitas innecesarias al supermercado de modo que se ahorra tiempo.

En esta aplicación, además se ha integrado un sistema de recompensa de puntos por tarea realizada. Para comprender que nos motiva a la hora de realizar una tarea habrá que determinar en qué consiste la motivación.

La motivación se puede definir como el proceso que inicia, guía y mantiene las conductas orientadas a lograr un objetivo o a satisfacer una necesidad.

Es un principio básico en la conducta de los seres humanos: nadie se mueve sin una razón para ello. Estar motivado significa llevar a cabo las tareas diarias sin que supongan una carga pesada y nos mantiene vivos. Pero no sólo eso, la motivación tiene relación con otras variables psicológicas, como el nivel de estrés, la autoestima, la concentración, etc., y tiene un efecto en la salud y el bienestar de todos nosotros.

Existen dos tipos de motivación, la motivación intrínseca que se refiere a aquella que emana de los propios individuos, es decir, hace referencia a la motivación que viene del interior del individuo más que de cualquier recompensa externa. Se asocia a los deseos de autorrealización y crecimiento personal.

El otro tipo de motivación que existe es la extrínseca que es en la que nos vamos a basar para conseguir los objetivos de realización de tareas a través de la asignación de puntos. La motivación extrínseca hace referencia a que los estímulos motivacionales vienen de fuera del individuo y del exterior de la actividad. Por tanto, los factores motivadores son recompensas externas.

La motivación extrínseca no se fundamenta en la satisfacción de realizar la cadena de acciones que compone aquello que estamos haciendo, sino en una recompensa que solo está relacionada con esta de manera indirecta, como si fuese un subproducto.

También podemos hablar de otros tipos de motivación como son la motivación negativa y la positiva.

La motivación positiva se refiere al proceso por el cual un individuo inicia o mantiene adherido una conducta gracias a la obtención de una recompensa positiva, sea externa o interna (por el placer de la actividad)

Sin embargo, la motivación negativa hace referencia al proceso por el cual una persona inicia o se mantiene adherida a una conducta para evitar una consecuencia desagradable, tanto externa (castigo, humillación, etc.) o interna (evitar la sensación de frustración o fracaso).

Por tanto, nuestra Web de desempeño de tareas se apoyará en una motivación positiva y como hemos indicado anteriormente externa.

De tal manera, el usuario que vaya realizando las tareas programadas irá recibiendo puntos de manera que podrá competir con otros usuarios para alcanzar un mayor número de puntos que los demás usuarios, así por tarea realizada se adjudicarán 25 puntos.

En un posterior desarrollo de la Web se puede crear recompensas según se alcancen una cantidad determinada de puntos que podrían ser mejoras de la página Web como conseguir diferentes estilos de diseño, imágenes, etc.

Esto podría ser un desarrollo a posteriori de esta aplicación que me parece interesante explorar en un futuro.

En definitiva, la finalidad de esta Web es facilitar la organización de tareas, uso de una agenda accesible y que además con la implantación de los puntos por logros obtenidos que sirva como estímulo para la consecución de tareas, por tanto, puede ser útil para adolescentes cuyos padres quieran conseguir que se motiven a la hora de realizar deberes, asistir a eventos o para profesionales que requieran administrar bien su tiempo tanto de trabajo como de ocio.

A la hora de planificar la realización de este proyecto se elaboró un procedimiento a seguir:

- Pensar en la necesidad de una web enfocada a la organización de tareas
- Recabar las necesidades que debe cumplir.
- Planteamiento de diseño o prototipo
- Elaborar unos Objetivos
- Elaborar la plataforma en sí



## 4. Requisitos y objetivos

### 4.1 Requisitos

La aplicación web Tasker nace de la necesidad de unir varias herramientas para ayudar a las personas a organizarse en su día a día. Entre estas herramientas se encuentran:

- Notas personalizadas.
- Tareas personalizadas.
- Lista de la compra.
- Un calendario.

En el apartado de notas el usuario podrá crear una nueva nota y visualizar notas ya creadas, así como eliminarlas, si fuera necesario.

En el apartado de tareas, el usuario podrá crear tareas indicando el día, hora y el texto que desee introducir. Una vez creadas, estas tareas podrán ser visualizadas además de ser marcadas como finalizadas, al hacerlo, el usuario recibirá 25 puntos que serán almacenados en la base de datos, esto es una forma de incentivar al usuario a completar sus tareas ya que al visualizar sus puntos obtendrá un feedback para las tareas que completa. Las tareas que no hayan sido completadas, es decir, ha pasado el tiempo para el que estaban programadas, serán eliminadas automáticamente y el usuario no recibirá puntos.

En el apartado de lista de la compra, el usuario podrá crear una nueva lista de la compra y visualizarla. En la elaboración de dicha lista, el usuario podrá añadir sus propios productos además de usar algunos proporcionados por la aplicación. Una vez creada la lista, los productos podrán ser marcados una vez comprados., lo que facilitará al usuario de la Web el momento de ir a comprar, así como posibles olvidos y se gestionará el tiempo de manera más eficaz.

En el apartado de calendario el usuario podrá ir cambiando de mes además de ver los días para los que tiene una tarea pendiente.

Para poder acceder a este contenido el usuario deberá hacer pinchar en login en la página principal. Si el usuario no tiene una cuenta podrá registrarse creando una nueva en el apartado Create account.

El usuario podrá recuperar su contraseña en el apartado Forgot password, de esta manera, el usuario recibirá un email con su nueva contraseña para recuperar su cuenta.

## 4.2 Objetivos

Tras hacer una valoración de las necesidades existentes a la hora de organizar las tareas que hay que desempeñar en los diferentes horarios y tras investigar las páginas web existentes, consideré desarrollar esta página para dar respuesta a estas necesidades, a continuación, se describen los objetivos del proyecto en detalle:

- Crear una aplicación visualmente atractiva e intuitiva para los usuarios.
- Elaborar un apartado de tareas personalizadas.
- Generar un editor de notas como una manera de apuntar rápida.
- Confeccionar una lista de la compra.
- Crear un calendario en el que se visualicen las tareas guardadas.
- Tener lista una web funcional para todo tipo de dispositivos.
- Proporcionar una aplicación web accesible para todos

## 5. Viabilidad y soluciones escogidas

### 5.1 La web

Durante este curso he estado trabajando con diferentes lenguajes de programación HTML, CSS, PHP, JavaScript, haciendo uso de técnicas como AJAX, programando usando el patrón MVC, subir proyectos a GitHub e incluso como crear prototipos y mapas de navegación.

He querido aplicar todos estos conocimientos adquiridos a este proyecto como una forma de implementar lo aprendido a lo largo de mi formación académica.

He decidido trabajar usando el entorno de desarrollo XAMPP ya que de esta manera puedo hacer uso del lenguaje PHP y una base de datos en PHPmyadmin permitiéndome usar HTML y dar funcionalidad a los elementos con JavaScript.

El código de la aplicación ha sido desarrollado en Visual Studio Code, un programa muy empleado para desarrollo web y que también he podido utilizar en las prácticas de la empresa.

### 5.2 Estudio preliminar SEO (Search Engine Optimization)

El SEO es necesario porque hoy en día no basta con tener un sitio web online posiblemente existen millones de webs similares y todas compiten por la atención del usuario. Para navegar por internet los buscadores se han convertido en una herramienta imprescindible y el buscador principal es Google. Éste ordena las webs utilizando unos criterios y algoritmos más o menos conocidos en base a la práctica.

Es importante tener en cuenta que una web que esté bien optimizada (código JavaScript, PHP y lenguaje de marcado claro y que se ejecute de forma eficiente) va a tener más “puntuación” en el baremo de Google, esto antes se llamaba Google Rank, pero ya no existe como tal propiamente dicho. Ahora simplemente funciona mediante la combinación de muchos algoritmos y evaluación de trabajadores.

A la hora de diseñar una web no solamente se debe tener en cuenta los requisitos técnicos como la tecnología del servidor, lenguajes de programación utilizados etc. También debemos tener en cuenta el contenido de la página. Este contenido debe ofrecer información clara y bien estructurada al usuario (eso mejora nuestra puntuación de Google). Por todo ello nuestro código de HTML debe estar bien estructurado.

Debe tener meta etiquetas: las más importantes son el título, la descripción y las palabras clave (<title>, <description> y <keywords>).



Figura 5.2.: Fuente: <https://www.alhsis.com/wp-content/uploads/2018/04/Meta-Tags-500x250.jpg>

Todas deben guardar una coherencia, por ejemplo, título: “Tasker”. Contiene las palabras “organize”, “tasks”, “notes”, “calendar”, “shopping list”. Si aparece repetido (pero con fundamento, si se repite contenido sin sentido esto nos penaliza) en títulos, encabezados, alts y cuerpo de texto, google nos sube la puntuación y nos posiciona bien. También existen otros factores más técnicos como la velocidad de carga del servidor.

## 5.3 Planificación de tareas

La planificación de tareas para el proyecto fue la siguiente:

- 1ª Fecha (1 – 27 abril) Especificación de requisitos y desarrollo de la base de datos:
  - Descripción del proyecto
  - Requisitos
  - Esquema E/R
  - Casos de uso
  - Desarrollo de la base de datos
- 2ª Fecha (2 – 11 mayo) Implementación de funcionalidades:
  - Eliminar notas
  - Marcar tarea como finalizada
  - Añadir los puntos al usuario al marcar tarea como finalizada
  - Eliminar productos de la lista de la compra
  - Visualizar las tareas programadas para la fecha correspondiente en el calendario
- 3ª Fecha (11 – 25 mayo) Implementación de funcionalidades y documentación
  - Validación de los casos de uso
  - Implementación de funcionalidades extras
  - Memoria
- 4ª Fecha (25 – 31 mayo) Finalización del proyecto:
  - Comprobaciones en el código
  - Repaso de la documentación

## 6. Diseño

### 6.1 Maqueta

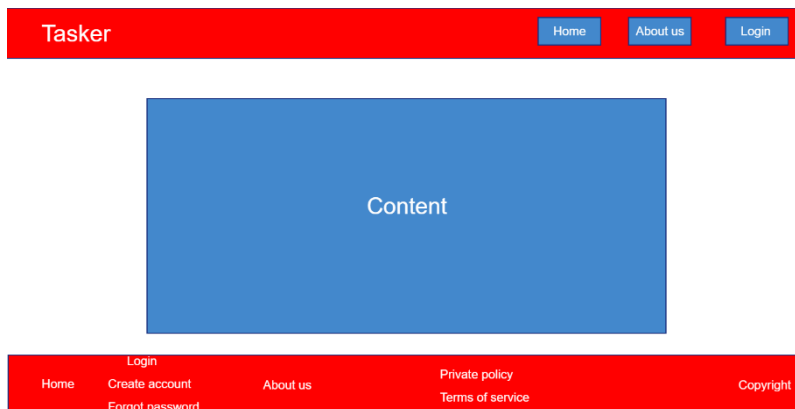
Un mockup o maqueta es un diseño digital de una web y / o aplicación. Las maquetas se utilizan en la fase de diseño inicial para visualizar ideas y conceptos en el contexto del diseño web e incluyen la estructura de navegación, el sitio y los elementos de diseño.

Para realizar las maquetas he utilizado el programa Pencil con el objetivo de tener una idea de la estructura antes de empezar a desarrollar la página.

Página de inicio:

En la primera imagen se observa una maqueta del sitio de inicio, hay un header con el logo y tres enlaces, en el centro se visualiza el contenido de la página y al final de esta hay un footer con varios enlaces además del copyright.

En la segunda imagen se aprecia la misma pantalla, pero para móviles. Se observa que ya no hay footer y que se accede a todos los enlaces mediante un menú situado en el header.



6.1.1 (izquierda): Página de inicio – PC



Figura

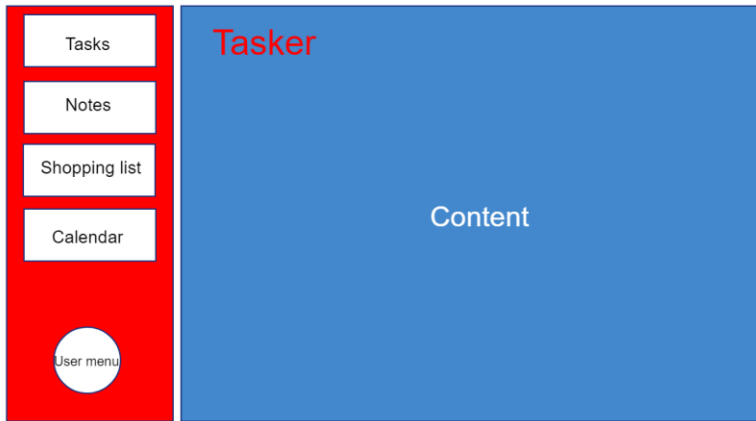
Figura 6.1.2 (derecha): Página inicio - móvil

### Aplicación web:

Las siguientes maquetas corresponden a la aplicación una vez el usuario inicia sesión. En la primera imagen se observa un menú lateral con varias secciones de la aplicación que al ser clicadas se cargaría su contenido en el centro de la página. También hay un header en la parte superior con logo y un menú desplegable con forma circular desde donde el usuario podrá cerrar sesión. Como se aprecia en las maquetas este menú circular podría haber aparecido en la parte de abajo del menú lateral pero finalmente se optó por la primera forma.

En la maqueta para móvil el menú lateral desaparece para ser accedido desde el menú de usuario.





**Figura 6.1.3 (arriba izquierda):** Página de la aplicación - PC

**Figura 6.1.3 (arriba derecha):** Página de la aplicación - móvil

**Figura 6.1.3 (arriba abajo):** Página de la aplicación v2 - PC

## 6.2 Estética del sitio

Una página web primero tiene que funcionar, pero luego ha de ser atractiva para el usuario. Para que una página web funcione correctamente y logre los objetivos planteados a la hora de su desarrollo hay que conseguir el equilibrio entre diseño y usabilidad.

El diseño es parte fundamental de la sociedad, en la actualidad nos centramos más en el diseño de un producto, si este diseño es adecuado, los resultados serán favorables, pero si un diseño no es atractivo, los resultados no serán buenos. Hay que tener conciencia de la importancia y del efecto que supone el diseño en nuestra vida diaria, y cómo el diseño ha sufrido transformaciones para adecuarse a las necesidades y deseos de las diferentes generaciones. El diseño es un proceso de creación visual con un propósito, ha de cubrir exigencias prácticas, debe transportar un mensaje prefijado.

Por tanto, la estética de una página web es importante ya que la primera impresión es fundamental cuando un cliente o usuario accede por primera vez a nuestra página o se interesa por nuestro proyecto y debe encontrar un diseño agradable. En esto reside la importancia de un sitio trabajado desde lo estético, bien diseñado, con una tipografía correctamente equilibrada y unos colores corporativos acordes.

Por tanto, hay que dedicar la atención que se merece al diseño de una página Web. Entre las ventajas de realizar un buen diseño en la presencia online podemos señalar:

- Aumenta la usabilidad de la página web: un buen diseño persigue una mejor experiencia del usuario durante la navegación, esto se traduce en mayor tiempo de visita en la web.

- Supone confianza a la hora de adquirir un producto: cuando un usuario quiere obtener un producto lo hará con mayor probabilidad en un entorno que le haga sentir bien.
- Aumenta el número de conversiones: si el usuario se siente cómodo permanece en el sitio. La calidad del diseño crea también una percepción positiva de la página que está utilizando.
- Genera más visitas a la página: A la vez también el número de recomendaciones y, por lo tanto, las visitas totales van a aumentar. Un buen diseño crea confianza a la hora de compartir contenidos.

Cuando nos referimos al diseño de una página web, no nos centramos exclusivamente en la estructura y distribución de elementos internos sino también a la gama de colores empleada, el tipo de estructura utilizado, los logos elaborados, el tipo de letra etc., es decir, en todo el conjunto.

Las tipografías para web actualmente son muy variadas, en mi proyecto la fuente utilizada es Louis-George-Cafe-Italic y los colores principales son el blanco, rojo, negro y gris. Es fundamental tener presente que cuando se realiza un diseño para un medio online en referencia con el texto hay que valorar también el contraste, el color, la legibilidad y el tamaño. Además, la teoría del color y la elección del color juegan un papel sumamente importante en la tipografía web. Con la estética que he elegido y elaborado en el proyecto he querido transmitir y reflejar profesionalidad.

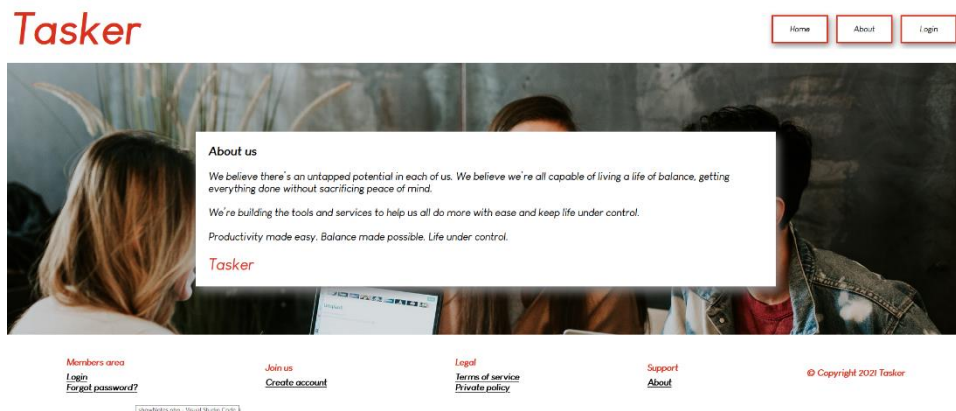


Figura 6.2.1: Página de inicio

En la página existen imágenes e iconos que a veces rompen la composición de colores, con el objetivo de atraer la vista del usuario haciendo la página más intuitiva y accesible además de quebrantar la monotonía cromática. Por ejemplo, la imagen para el menú de usuario en la parte superior izquierda o los iconos para marcar las tareas como finalizadas.



Figura 6.2.2: Menú de usuario

## Your tasks

### Today

play football at 21:03



### Tomorrow

go to the doctor at 12:30



### Someday

Figura 6.2.3: Tareas

## 6.3 Mapa del sitio

El mapa de navegación para la página de acceso a la aplicación es:

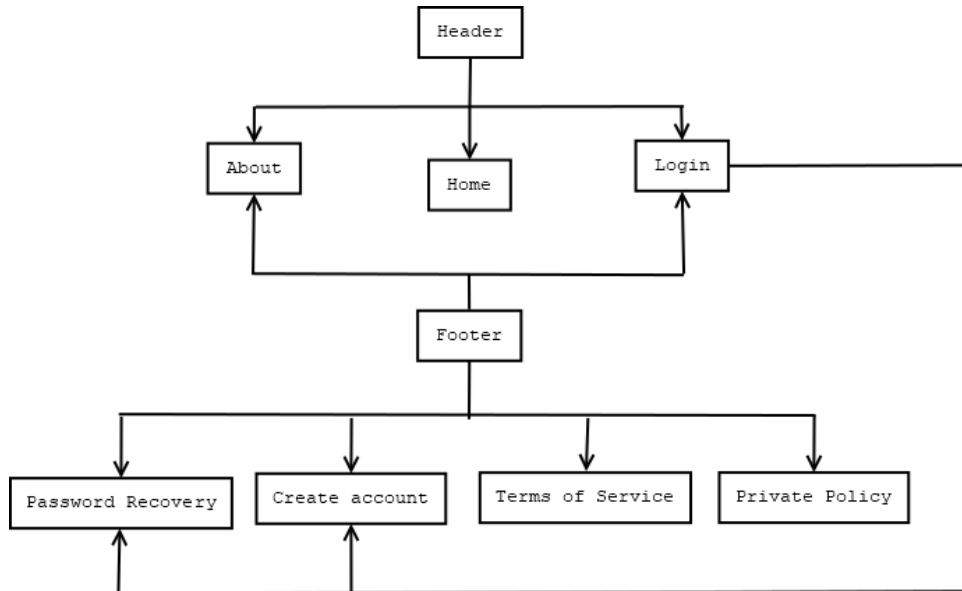


Figura 6.3.1: Mapa de navegación página de inicio



El mapa de navegación para la aplicación en sí, es decir una vez el usuario ha iniciado sesión es:

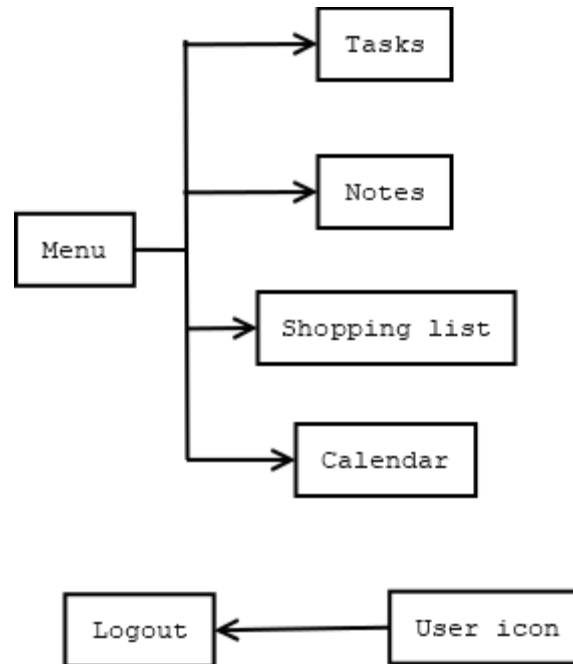


Figura 6.3.2: Mapa de navegación página de la aplicación

## 6.4 Modelo de entidad-relación

Un modelo entidad-relación es una herramienta para el modelo de datos, la cual facilita la representación de entidades de una base de datos. Las entidades se representan con sus atributos, diferenciando entre las claves primarias, atributos multivalor, derivados, etc.

También se representan las relaciones entre las entidades indicando la cardinalidad. Por ejemplo: un usuario puede crear de una a indefinidas notas, pero una nota solo puede ser creada por un usuario, esto indica que la relación es de 1 a N.

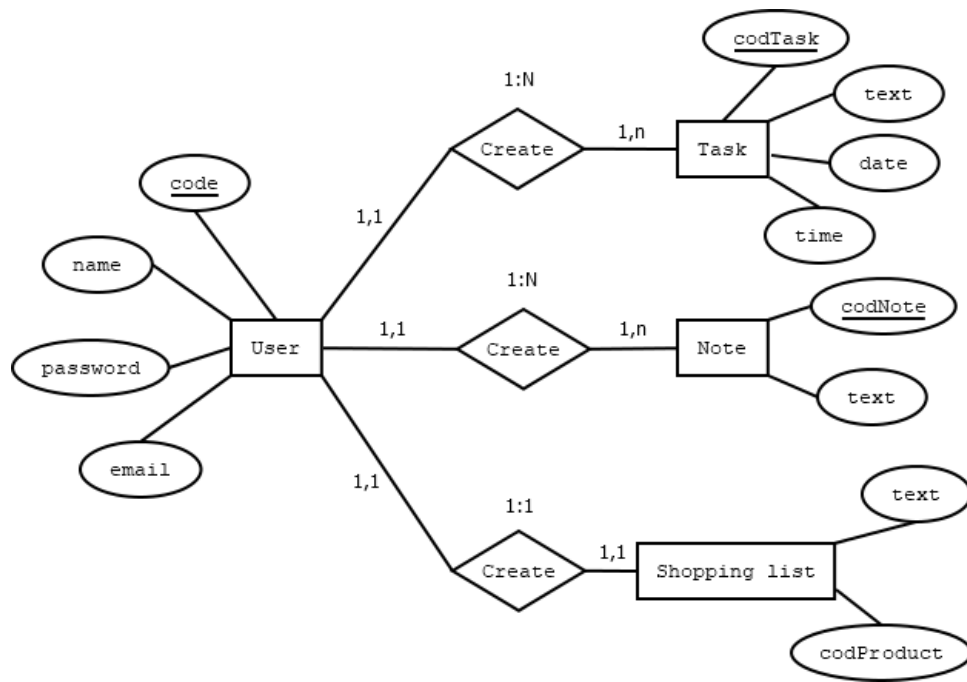


Figura 6.4: Modelo de E/R

## 6.5 Diagrama de casos de uso

Los diagramas de casos de uso son una forma de diagrama de comportamiento UML mejorado. En ellos se incluyen las acciones que puede realizar un usuario llamado actor en una página web.

En el siguiente diagrama se pueden apreciar algunas relaciones <<include>>. Las inclusiones se producen cuando un caso de uso base incorpora explícitamente el comportamiento de otro en algún lugar de su secuencia.

La relación de inclusión sirve para enriquecer un caso de uso con otro y compartir una funcionalidad común entre varios casos de uso.

Por ejemplo: crear una tarea incluye la acción de crear un texto y esto también se hace al crear una nota.

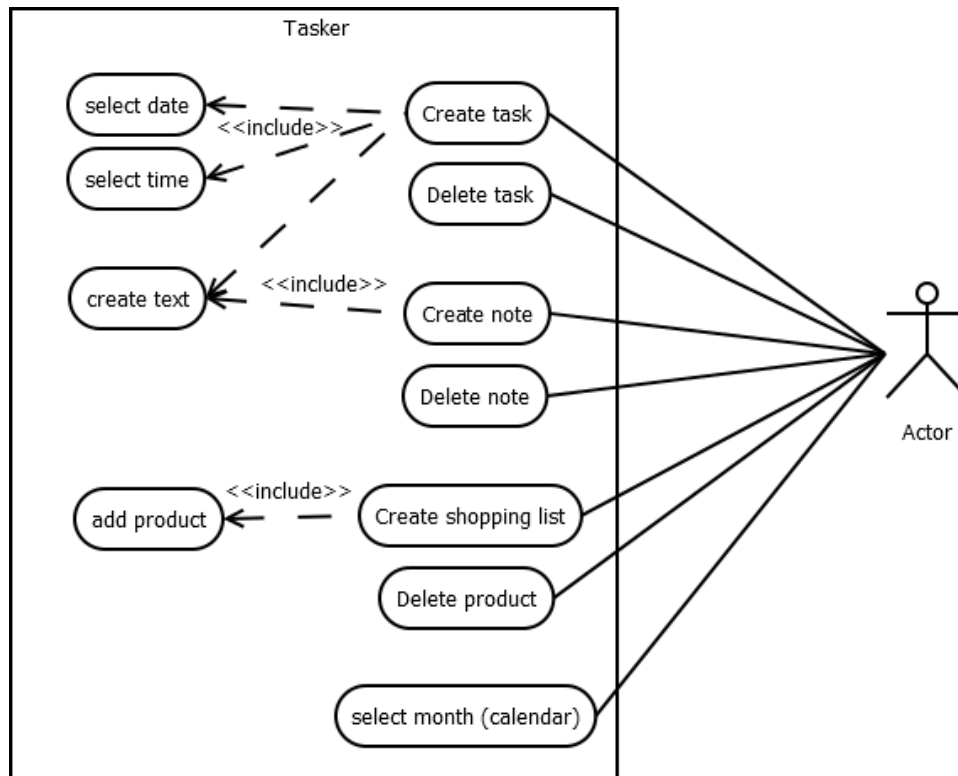


Figura 6.5: Diagrama de casos de uso

## 6.6 Modificaciones y aprobación de diseño

El prototipo, como boceto de un concepto está sujeto a cambios durante el desarrollo. Es muy extraño que una web termine siendo exactamente igual que un prototipo, a no ser que el cliente tenga una idea muy cerrada de cómo quiere que sea su web y no admita cambios. Lo habitual es que durante el desarrollo vayan surgiendo dificultades de tipo técnico o que se nos planteen mejoras que hasta el momento no se nos habían ocurrido.

Pese a esto, en mi caso la mayoría de mejoras y cambios en la página no son desde el punto de vista estético, sino del funcional. Los cambios más notorios en la versión final con respecto a la maqueta son:

- Para la página de inicio en formato móvil se ha mantenido el footer pero con los elementos en sentido vertical.
- En la página de la aplicación no hay header, no hay logo de la página, pero el menú de usuario sigue presente en la parte superior derecha.
- En la página de la aplicación en formato móvil el menú lateral se ha movido a la zona superior de la página como un footer en lugar de eliminarlo y poner los enlaces dentro del menú de usuario.

A lo largo de estos meses de trabajo he ido introduciendo nuevos elementos o acciones en base a las sugerencias de Xavier, el coordinador del proyecto, testers o simplemente mejoras que se me ocurrían.

## 7. Herramientas

### 7.1 Servidor XAMPP

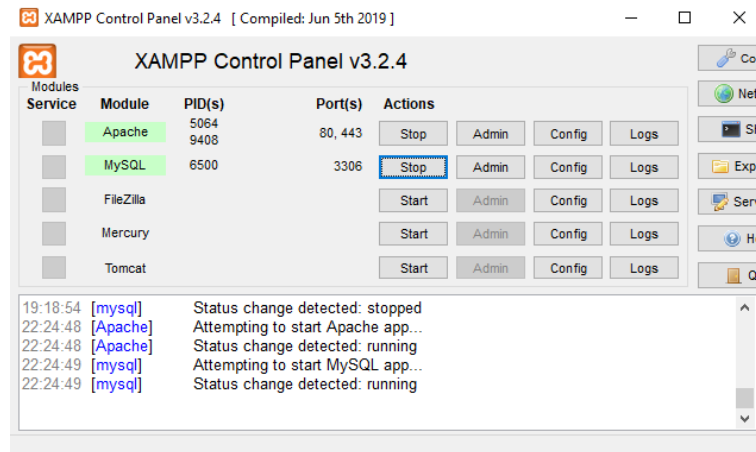


Figura 7.1.1: XAMPP Control Panel

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Se puede descargar desde la página:

<https://www.apachefriends.org/es/index.html>

En el momento en el que tendremos que poner mayor atención es a la hora de instalar MySQL (motor de la base de datos) y PHPMyAdmin (gestor gráfico de base de datos).

Para acceder a nuestro servidor local, tendremos que arrancar los servicios de Apache y MySQL e introducir en nuestro navegador la dirección "localhost". Tendremos que colocar la carpeta "dt\_final\_project" que contiene los subdirectorios de la web en la carpeta "htdocs". Una vez hecho esto la página podrá visualizarse en el navegador mediante "localhost/dt\_final\_project".

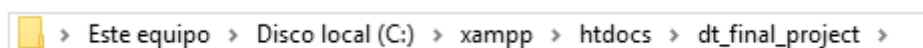


Figura 7.1.2: Dirección local proyecto

Para acceder a PHPMyAdmin la dirección del navegador es "localhost/phpmyadmin/". Desde aquí creamos la tabla que recogerá los datos del formulario.

Para crear la tabla habrá que clicar en el menú de la izquierda y pulsar nueva. A continuación, se introducirá el nombre de la base de datos: “dt\_final\_project” y habrá que pulsar el botón Crear.

Una vez creada, en el menú de arriba se selecciona importar y desde esa pestaña buscaremos el archivo “dt\_final\_project.sql” que se encuentra en la carpeta de dentro de la carpeta donde se adjunta el proyecto.

La versión empleada en el proyecto es la 3.2.4.

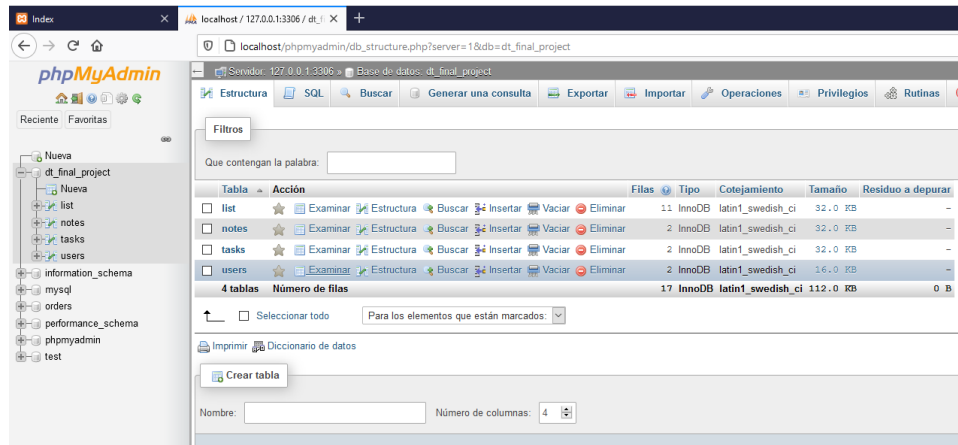


Figura 7.1.3: Vista base de datos desde PHPMyAdmin

## 7.2 PHP



Figura 7.2: PHP Logo. Fuente: <https://nestrategia.com/wp-content/uploads/2019/04/php-desarrollo-web.png>

PHP es un acrónimo recursivo que significa Hypertext Pre Processor (originariamente PHP Tools o Personal Home Page Tools). Creado por Rasmus Lerdorf en 1994 es, sin embargo, su implementación realizada actualmente por The PHP Group y se utiliza como el estándar de hecho para PHP al no existir una especificación formal. Publicado bajo la licencia PHP License, la Free Software Foundation considera esta licencia como software libre.

Es un lenguaje de programación interpretado, diseñado inicialmente para la elaboración de páginas Web dinámicas. Es utilizado sobre todo en interpretación del lado del servidor (server-side-scripting) pero en la actualidad podemos utilizarlo desde una interfaz de línea de comandos o en la creación de otra clase de

programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK.

La versión de PHP empleada en el proyecto es la 8.0.2.

### 7.3 JavaScript



**Figura 7.3:** JavaScript Logo. Fuente: <https://www.cloudstudio.mx/blog/wp-content/uploads/2019/01/js.png>

Es un lenguaje de programación interpretado, es decir, que no requiere compilación. Se utiliza sobre todo en páginas Web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

Como el lenguaje Java, el JavaScript, está enfocado a objetos propiamente dicho, ya que dispone de Herencia, si bien ésta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipo) y extendiendo su funcionalidad.

Prácticamente todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas Web.

Para interactuar con una página Web se provee al lenguaje Java Script de una implementación del DOM.

### 7.4 AJAX



**Figura 7.4:** Ajax Logo. Fuente: [https://cdn.goconqr.com/uploads/flash\\_card/image\\_question/9661593/desktop\\_cb025d9d-c015-407e-83d2-dd97c4460b0b.png](https://cdn.goconqr.com/uploads/flash_card/image_question/9661593/desktop_cb025d9d-c015-407e-83d2-dd97c4460b0b.png)

Es un acrónimo de Asynchronous JavaScript And XML, consiste en una técnica de desarrollo Web para crear aplicaciones interactivas o RIA (Rich Internet Applications).

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De manera que es factible realizar modificaciones sobre las paginas sin necesidad de recargarlas, lo cual implica aumentar la interactividad, velocidad y un mayor uso de las aplicaciones.

Es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni en el comportamiento de la página. JavaScript es el lenguaje interpretado en el que se realizan normalmente las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, que está disponible en los navegadores actuales. De cualquier forma, no es necesario que el contenido asíncrono esté formateado en XML.

AJAX es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores ya se basa en estándares abiertos como JavaScript y Document Object Model(DOM).

## 7.5 SQL



**Figura 7.5:** SQL Logo. Fuente: [https://miro.medium.com/max/652/0\\*T6N1\\_5m6-H2k3bKN](https://miro.medium.com/max/652/0*T6N1_5m6-H2k3bKN)

El SQL Structured Query Language (traducido como lenguaje de consulta estructurado) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Entre sus principales características está el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el objetivo de recuperar de manera sencilla información de interés de una base de datos, así como también poder realizar modificaciones sobre ella. Es un lenguaje de cuarta generación (4GL).

La versión de MySQL empleada en el proyecto es la 5.7, esta se instala con XAMPP.

## 7.6 HTML



**Figura 7.6:** HTML Logo. Fuente: <https://www.luisan.net/disenio-paginas-web/imagenes-desarrollo-web-tecnologia/html5-logo.jpg>

Es un lenguaje de diseño cuyas siglas son un acrónimo de HyperText Markup Language (traducido del inglés como Lenguaje de Marcas de Hipertexto), es el lenguaje de marcado predominante para la construcción de páginas Web.

Se utiliza para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTML se escribe en forma de "etiquetas", rodeadas por corchetes angulares (). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

HTML también es utilizado para referirse al contenido del tipo de MIME text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores). Por convencionalismo, los archivos de formato HTML usan la extensión .htm o .html.

## 7.7 CSS



**Figura 7.7:** CSS Logo. Fuente: [https://desarrolloweb.com/storage/tag\\_images/actual/ST1RLpDHZlnATuKnDUkwXhKoalOrtS97gBtgiQ6M.png](https://desarrolloweb.com/storage/tag_images/actual/ST1RLpDHZlnATuKnDUkwXhKoalOrtS97gBtgiQ6M.png)

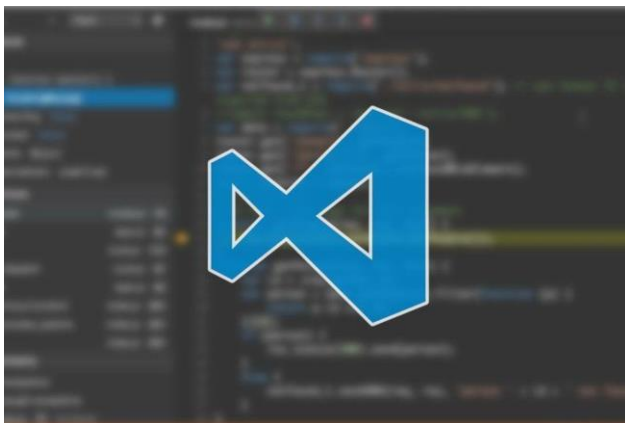


CSS es un lenguaje de diseño este término viene del acrónimo en ingles Cascading Style Sheets, (cuya traducción al español seria Las hojas de estilo en cascada).

CSS es un lenguaje artificial utilizado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

## 7.8 VISUAL STUDIO CODE



**Figura 7.8:** Visual Studio Code Logo. Fuente: <https://live.mrf.io/statics/i/ps/www.muylinux.com/wp-content/uploads/2015/11/visual-studio-code.jpg?width=1200&enable=upscale>

Visual Studio Code (Traducido como código de estudio visual) es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS, permite trabajar con diversos lenguajes de programación.

Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

Se caracteriza sobre todo porque puede ser personalizable, permite gestionar tus propios atajos de teclado y refactorizar el código, también nos proporciona gran utilidad para descargar y gestionar extensiones con las que podemos personalizar y potenciar esta herramienta. Las extensiones de Visual Studio Code nos otorgan infinidad de opciones, como colorear tabulaciones, etiquetas o recomendaciones de autocompletado.

También hay extensiones que nos ayudan con el lenguaje de programación que vayamos a usar, como por ejemplo para Python, C / C++, JavaScript, etc.

Sin embargo, muchas de las características de Visual Studio Code no están expuestas a través de los menús o la interfaz de usuario. Más bien, se accede a través de la paleta de comandos o a través de archivos. json (por ejemplo, preferencias del usuario).

Es gratuito, de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.<sup>3</sup>

La versión de Visual Studio Code empleada para realizar el proyecto es la 1.56.2.

## 7.9 PENCIL



**Figura 7.9:** Pencil Logo. Fuente: <https://atalantic.com/wp-content/uploads/2017/05/pencil.jpg>

Es una herramienta que permite realizar prototipos para aplicaciones y sitios web de código abierto, multiplataforma, con extensión para Firefox.

Contiene las plantillas para diagramas y prototipos, edición de texto en pantalla con texto enriquecido, así como las operaciones de dibujo estándar.

Además, los archivos preliminares o definitivos, pueden exportarse en distintos formatos, PDF, PNG, SVG u ODT. Aunque inicialmente solo trabajaba con Mozilla Firefox, ahora soporta todas las plataformas.

Pencil Project está pensada directamente para los desarrolladores y diseñadores de aplicaciones de escritorio y web que quieran tener un modelo de su proyecto antes de ponerse a construirlo.

La aplicación nos va a proporcionar una forma muy sencilla y rápida de crear un diseño que mostrar a un cliente antes de meternos a fondo en su desarrollo y programación. El proceso lo podremos llevar a cabo arrastrando y soltando.

También podremos hacer uso de las colecciones de formas, lo cual simplifica considerablemente el diseño.

Ofrece varias colecciones de formas para dibujar diferentes tipos de interfaz de usuario que van desde aplicaciones de escritorio hasta plataformas móviles.

La versión empleada para realizar la maqueta es la 3.1.0.

## 7.10 PHP Mailer

PHPMailer es una clase creada específicamente para hacer sencillo el envío de emails que tienen características complejas mediante PHP, es bastante útil y potente. Permite enviar emails con archivos adjuntos, diferentes tipos de servidores SMTP (con o sin autenticación de usuario), a la vez que da soporte a otras características de la función mail() nativa de PHP, pero de una manera un poco más simple.

Está disponible con licencia de libre distribución y uso, código abierto.

No hace falta instalarlo ya que, en la carpeta del proyecto, es decir, “dt\_final\_project” ya se encuentra instalado en el subdirectorio “composer”.

En los archivos de la página hay que configurar las funciones newUser y checkMail del fichero connection.php. En concreto en las líneas donde pone “IMPORTANT” hay que cambiar el username, password y setFrom con los datos reales de un correo electrónico que se usaría como dirección para mandar los correos a los usuarios de la aplicación cuando se crean una cuenta nueva y cuando quieren recuperar su contraseña.

También es importante recordar que en estas dos funciones se emplea en el “require” el path del fichero “autoload.php” que ya está instalado en la carpeta “composer”, si se desea cambiar este path habría que hacerlo en estas líneas.

Otra cosa importante es que para que un usuario pueda recibir un email, deberá cambiar la configuración “Permitir aplicaciones menos seguras” en Gmail.

```
function newUser($newuser,$newkey,$newmail){
    $connection_string='mysql:dbname=dt_final_project;host=127.0.0.1';
    $u='root';
    $k='';
    try{
        require "../composer/vendor/autoload.php"; //IMPORTANT
        $mail = new PHPMailer();
        $mail->IsSMTP();
        $mail->SMTPDebug = 0;
        $mail->SMTPAuth = true;
        $mail->SMTPSecure = "tls";
        $mail->Host = "smtp.gmail.com";
        $mail->Port = 587;
        $mail->Username = "IMPORTANT";
        $mail->Password = "IMPORTANT";
        $mail->SetFrom('IMPORTANT', 'Tasker');
        $mail->Subject = "Welcome to Tasker";
        $mail->MsgHTML('Welcome '.$newuser.", now you can login and start planning");
        $mail->AddAddress($newmail, "Tasker");
        $x = $mail->Send();
        if(!$x) {
            echo "Error" . $mail->ErrorInfo;
        } else {
            echo "Sent";
        }
    }

    $bd=new PDO($connection_string,$u,$k);
    $ins="insert into users(name, passwd, mail) values ('$newuser', '".password_hash($newkey, PASSWORD_DEFAULT)."', '$newmail')";
    $resul=$bd->query($ins);
    return $resul;
} catch(PDOException $e){
    echo "Error in the database: ".$e->getMessage();
}
}
```

Figura 7.10: Código PHP Mailer.

Aunque para este proyecto no es necesario instalarlo debido a que todo lo necesario se encuentra en la carpeta “composer” para poder instalarlo desde cero hay que seguir los siguientes pasos:

- 1) Instalar Composer
- 2) Ejecutar el comando: `composer require phpmailer/phpmailer` desde la consola de comandos en la carpeta en la que lo queramos instalar
- 3) En esa carpeta se generará un archivo “autoload.php” cuyo path usaremos mediante la etiqueta “require” en los ficheros donde se emplee PHPMailer

En la siguiente página hay un tutorial muy recomendable para instalarlo:

<https://alexwebdevelop.com/phpmailer-tutorial/>

La versión de Composer empleada en este proyecto es la 2.0.11.

## 7.11 Repositorio GitHub



**Figura 7.11:** GitHub Logo. Fuente: <http://blogs.encamina.com/por-una-nube-sostenible/wp-content/uploads/sites/19/2021/03/github.png>

He utilizado Github como espacio para guardar mi proyecto de manera que el coordinador del proyecto pudiera acceder al mismo, revisarlo además de evaluar y poder comprobar todos los cambios que introducía en tiempo real.

Github es un repositorio de software y de control de versiones. Funciona mediante un entorno gráfico o consola. Este año he utilizado la aplicación Github Desktop para Windows que en mi opinión es más sencilla de usar porque puedes ver de forma más clara los cambios en el código.

Para usarlo tenemos que crear una cuenta y activarla. Después instalamos el software en el PC y ya lo tenemos listo. Se empieza “clonando” el repositorio después de crearlo en nuestro perfil, o al revés, podemos crear un repositorio y subirlo.

El procedimiento para subir código que he empleado ha sido copiar los archivos modificados al repositorio local y subir los cambios con la aplicación de Github Desktop.

Esta metodología de trabajo no es la más eficiente porque hay que ir copiando y pegando los archivos que hayas editado al repositorio, pero como el

proyecto lo empecé a desarrollar un tiempo antes de realizar las prácticas en la empresa aún no había conocido la forma de integrar git con Visual Studio Code de forma que los archivos los modificas y los subes a remoto desde ahí.

La versión de GitHub desktop empleada en este proyecto ha sido la 2.8.2 para Windows x64.

## 8. Base de datos

En este apartado se explicarán los datos almacenados en la base de datos de este proyecto, así como los datos de ejemplo. En el apartado 7.1 de esta memoria se explica cómo importar la base de datos.

### 8.1 Diagrama

El siguiente diagrama muestra las tablas que componen la base de datos y como se relacionan entre sí. Se puede observar desde la pestaña Diseñador del menú superior en localhost/phpmyadmin cuando clicamos en la base de datos.

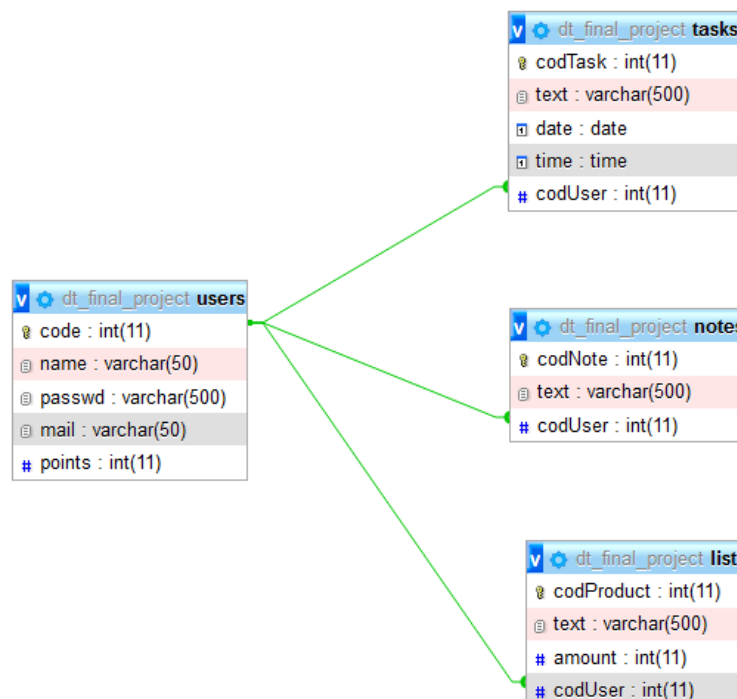


Figura 8.1: Diagrama de la base de datos

## 8.2 Almacenamiento

En la base de datos se almacenarán las siguientes tablas:

Users: tabla que guarda la información de los usuarios de la aplicación

- Code: Integer (11) autoincrementado. Es la clave primaria, cada usuario tendrá un código diferente al resto. Se usa para poder diferenciar los usuarios.
- Name: Varchar (50). Es el nombre de usuario que se emplea para iniciar sesión y se muestra dentro de la aplicación.
- Passwd: Varchar (500). Es la contraseña de usuario, está codificada en hash.
- Mail: Varchar (50). Es el email del usuario en el que recibirá correos de la aplicación.
- Points: Integer (11). Son los puntos que tiene el usuario por haber completado tareas, por defecto tendrá 0.

Tasks: tabla que recoge las tareas creadas por el usuario.

- CodTask: Integer (11) autoincrementado. Es la clave primaria, cada tarea tendrá un código diferente al resto.
- Text: Varchar (500). Es el texto que el usuario escribe para una tarea, por ejemplo: ir a la compra.
- Date: Date. Es la fecha para la que el usuario programa la tarea.
- Time: Time. Es la hora para la que el usuario programa la tarea.
- CodUser: Integer (11). Es la clave foránea que relaciona la tarea con el usuario que la ha creado.

Notes: tabla que recoge las notas creadas por los usuarios.

- CodNote: Integer (11) autoincrementado. Es la clave primaria, cada nota tendrá un código diferente al resto.
- Text: Varchar (500). Es el texto que el usuario escribe para una nota.
- CodUser: Integer (11). Es la clave foránea que relaciona la nota con el usuario que la ha creado.

List: tabla que recoge los productos que el usuario ha añadido a su lista de la compra.

- CodProduct: Integer (11) autoincrementado. Es la clave primaria, cada producto tendrá un código diferente al resto.
- Text: Varchar (500). Es el nombre del producto.
- Amount: Integer (11). Es la cantidad de producto que el usuario selecciona, por ejemplo: 3 Manzanas.
- CodUser: Integer (11). Es la clave foránea que relaciona el producto con el usuario que ha creado la lista.

### 8.3 Datos de prueba

En el punto 7.1 se explica cómo importar la base de datos. Al hacer esto no solo se crean las tablas, sino que además se rellenan con datos que ya vienen por defecto. Estos datos de prueba son:

Para la tabla users:

code	name	passwd	mail	points
1	juan	1234	invented@gmail.com	250
3	maria	123	invented@gmail.com	0

Si se mira la tabla desde el navegador en localhost/phpmyadmin las contraseñas están encriptadas pero el valor real es el que aparece en esta tabla. Estas direcciones de correo no existen, si se quiere recibir correos se deberá cambiar estos correos o crear un nuevo usuario con un correo real.

Para la tabla tasks:

codTask	text	date	time	codUser
103	Play football	2021-04-30	23:46:00	1

Es importante recordar que la aplicación elimina las tareas si se ha pasado su fecha y tiempo, por lo tanto, el usuario no verá ninguna tarea al entrar en la aplicación.

Para la tabla notes:

codNote	text	codUser
1	I have to do the activities on page 98 for tomorrow	1
67	Films to watch: - Parasite - Gravity	1

Para la tabla list:

codProduct	text	amount	codUser
17041	milk	6	1
17042	cheese	2	1
17043	chicken	1	1
17044	bananas	4	1

## 9. Manual de usuario

### 9.1 Estructura de archivos

El proyecto está dividido en varias subcarpetas para facilitar la organización.

En la carpeta composer se encuentra todo lo necesario para utilizar los servicios que ofrece PHPMailer.

En la carpeta css se encuentran los archivos que dan estilo a los elementos HTML de la aplicación además de la fuente utilizada.

En la carpeta db se encuentra el archivo a importar en PHP para crear las tablas de la base de datos.

En la carpeta images se encuentran los iconos e imágenes empleadas en la página.

En la carpeta js se encuentran los archivos JavaScript que modifican el comportamiento de los elementos HTML.

En la carpeta pages se encuentran los archivos HTML y PHP que dan forma a la aplicación.

Por último, el archivo “index.html” es la página de inicio a la aplicación que se carga cuando introducimos en el navegador la dirección de la página.

### 9.2 Inicio

- Index.html

La página de inicio corresponde al archivo “index.html”. El código contiene los elementos HTML que se muestran al acceder, es decir, el contenido textual en el centro, un header y un footer con enlaces.

Se incluyen dos archivos css (uno interno y otro externo) y un archivo menuFunction.js que permite desplegar los enlaces del header al clicar en el menú que se genera al estar la página en formato móvil, esto se hace agregando una clase en el archivo JavaScript y mostrándolo con el css.

Cabe resaltar que, mientras index.html tiene un header y un footer en el propio código, el resto de archivos de esta sección de inicio importan header.html y footer.html y el correspondiente header-footer.css. Esto podría parecer mala práctica porque se está repitiendo el código, pero en realidad no lo es ya que index.html se encuentra en un directorio superior al resto de páginas y por tanto la ruta de los enlaces es distinta.



```
<!DOCTYPE html>
<html>
  <head>
    <title>Tasker</title>
    <meta charset="UTF-8" name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="The web application that will help you organize your life">
    <meta name="keywords" content="tasker, organize, tasks, notes, calendar, shopping list">
    <meta name="author" content="David Tobalina">
    <link href="css/index.css" rel="stylesheet" type="text/css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <script src="js/menuFunction.js"></script>
  </head>
  <body>
    <header>
      <h1>Tasker</h1>
      <nav id="links" class="topnav">
        <a href="javascript:void(0);" class="icon" onclick="myFunction()">
          <i class="fa fa-bars"></i>
        </a>
        <a href="index.html" class="home">Home</a>
        <a href="pages/about.php">About</a>
        <a href="pages/login.php">Login</a>
      </nav>
    </header>
    <section class="main">
      <article>
        <h2 class="titulo">Welcome to Tasker</h2>
        <p>The web application that will help you organize your life</p>
      </article>
      <article>
        <h3 class="titulo">Create tasks!!</h3>
        <p>You can visualize the created tasks and if you check them as done you will receive points!!</p>
        
      </article>
      <article>
        <h3 class="titulo">Create notes!!</h3>
        <p>You can create notes, visualize and delete them</p>
        
      </article>
      <article>
        <h3 class="titulo">Create your shopping list!!</h3>
        <p>Create your shopping list and cross out the products</p>
      </article>
    </section>
  </body>
</html>
```

Figura 9.2.1: index.html

```
function myFunction() {
  var x = document.getElementById("links");
  if (x.className === "topnav") {
    x.className += " responsive";
  } else {
    x.className = "topnav";
  }
}
```

Figura 9.2.2: menuFunction.js

```
@media screen and (max-width: 680px) {
  body {
    padding-top: 9rem;
  }
  #links.responsive {position: relative;}
  #links.responsive > .icon {
    float: right;
  }
  #links.responsive > a {
    float: none;
    display: block;
    text-align: left;
  }
}
```

Figura 9.2.3: index.css

- Terms.php y private.php

Los archivos terms.php y private.php accesibles desde el footer muestran un texto legal generado por las páginas: <https://www.termsofservicegenerator.net/> y <https://www.generateprivacypolicy.com/>

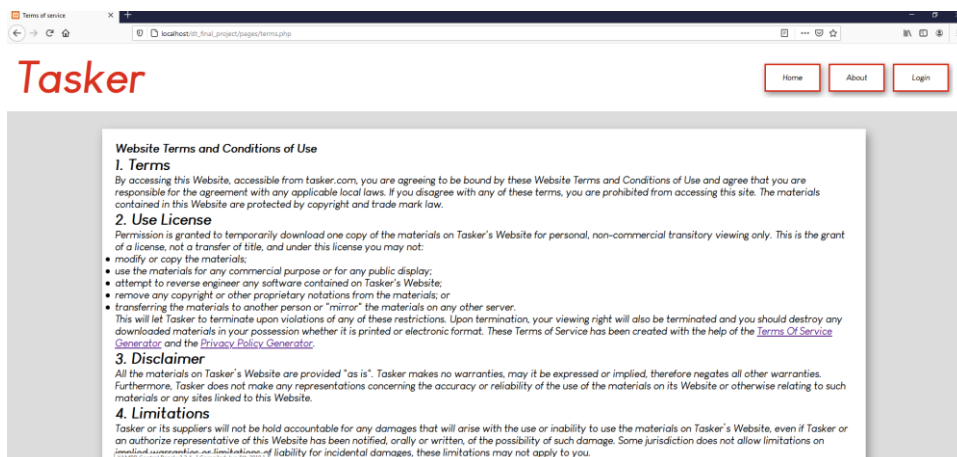


Figura 9.2.4: Vista de terms.php

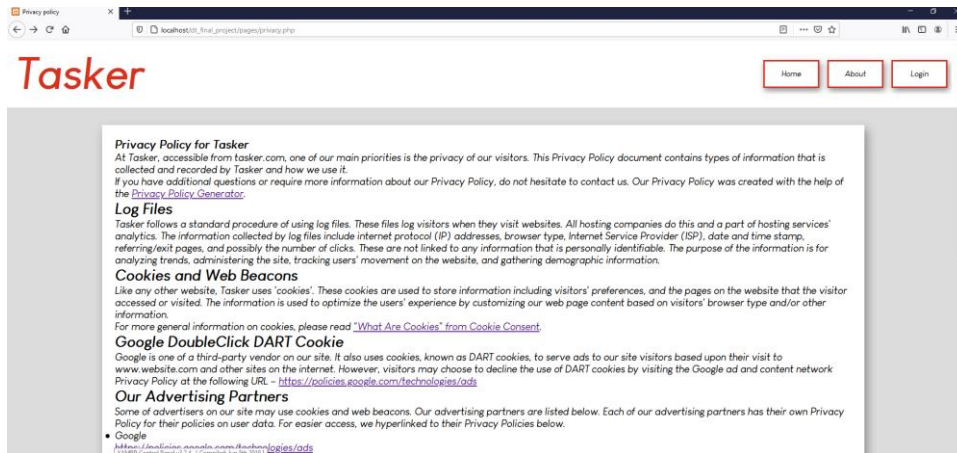


Figura 9.2.5: Vista de privacy.php

- About.php

El archivo about.php muestra un texto con una breve descripción de que es Tasker y los objetivos que se persiguen con su creación.

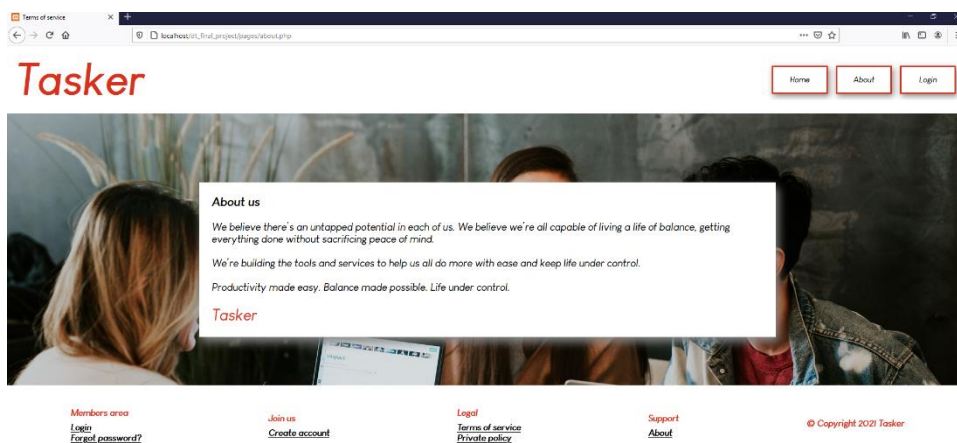


Figura 9.2.6: Vista de about.php

- Login.php

El archivo Login.php es un formulario en el que el usuario deberá introducir sus credenciales. Los datos se envían a este mismo archivo con el objetivo de validarlos antes de redirigir a la página principal.

En la línea 15 del código se comprueba mediante una función de connection.php que el usuario y la contraseña existen en la base de datos. Si no coinciden se muestra un error por pantalla, en caso contrario se redirige a main.php no sin antes ejecutar la función session\_start() de php que inicia una sesión y comprobar si el check de “Remember me” está activado, si lo está se crea una cookie que se destruye cuando el usuario hace logout. Si esta cookie no se destruye el usuario no tendrá que hacer login otra vez al acceder a main.php, se comprueba si la cookie existe en la línea 5.

Cabe destacar que en la línea 25 se crea una cookie de sesión con el nombre de usuario en su interior, esto permite utilizar el nombre de usuario dentro de la aplicación siempre que la sesión esté activa recogiendo el valor de esta cookie, lo cual será muy útil en los archivos de dentro de la aplicación para obtener el código del usuario, sacar el nombre por pantalla, etc.

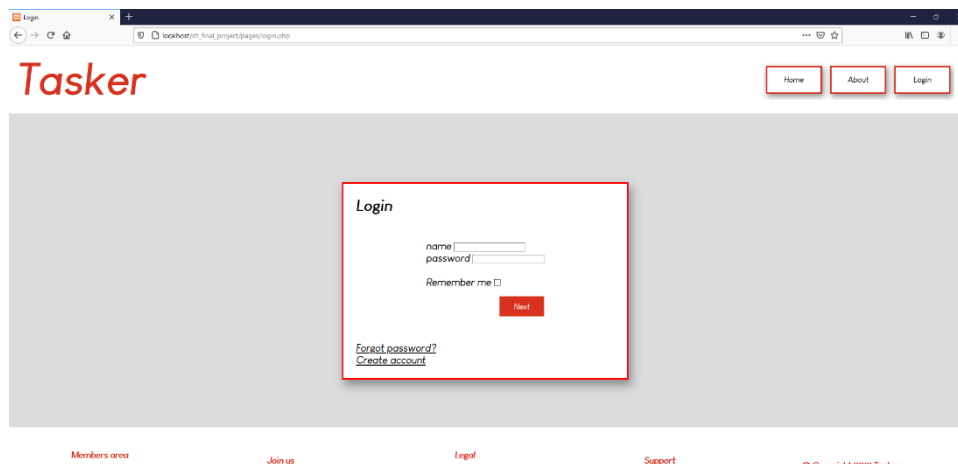


Figura 9.2.7: Vista de login.php

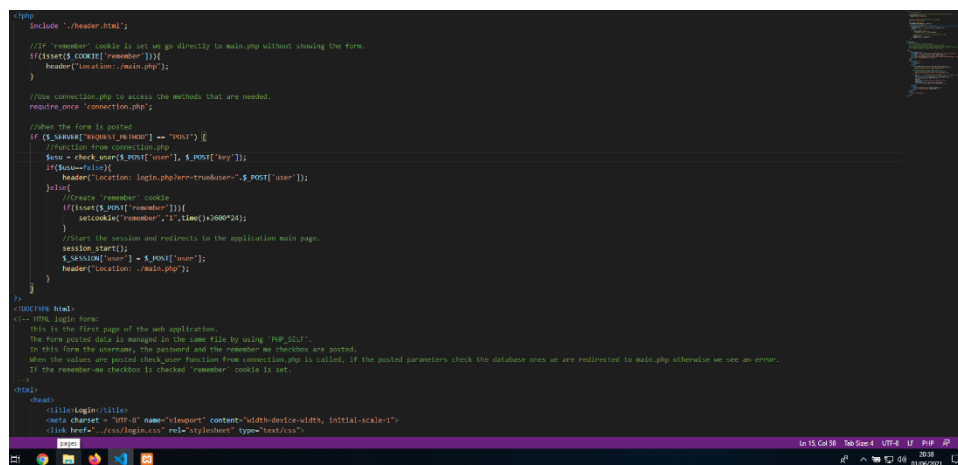


Figura 9.2.8: Código de login.php

- Sessions.php y logout.php

En el punto anterior se mencionaba que al hacer login se crea una sesión usando el método session\_start(), esto sirve para que cada usuario solo pueda ver sus archivos, es decir, si el usuario Juan ha escrito una nota no queremos que el usuario María pueda verla también. Y para comprobar que se ha logeado se crea una función en el archivo sessions.php que comprueba si en la cookie de \$\_SESSION que se crea al haber una sesión iniciada tiene al usuario almacenado, si no lo tiene se redirige a login.

```

1  <?php
2  /*This php part is a checking: if you try to access a file
3      function check_session(){
4          session_start();
5          if(!isset($_SESSION['user'])){
6              header("Location: login.php?redirected=true");
7          }
8      }

```

Figura 9.2.9: sessions.php

El archivo logout.php destruye todos los datos que genera una sesión, la cookie de “Remember me” que vimos en el punto anterior y otra cookie que veremos más adelante llamada “calendar”. Además, se redirige a index.html.

```

<?php
    //This is the logout file, it destroys the session
    //Join session
    session_start();
    //Empty session array
    $_SESSION = array();
    //Destroy the session
    session_destroy();
    //Delete the session cookie
    setcookie(session_name(), 123, time() - 1000);
    //Delete the remember me cookie
    setcookie("remember", "1",time() - 1000);
    //Delete the calendarcookie
    setcookie("calendar", "",time() - 1000);
    //Redirect to login
    header("Location: ../");

```

Figura 9.2.10: logout.php

- PasswdRecovery.php

Este archivo contiene un pequeño formulario con un solo campo que es el correo de un usuario. Los datos se envían al mismo archivo con el objetivo de comprobar los datos.

Cuando se envía el formulario se comprueba que el campo no esté vacío. Posteriormente se llama a la función checkMail() de connection.php que comprueba primero si el correo existe en la base de datos y luego si esto se cumple, se manda un correo a esta dirección haciendo uso de PHPMailer.

En el código de la función se observa un contador, este se usa para devolver un String en caso de que no se haya encontrado ningún usuario en la base de datos cuyo correo coincida con la dirección de proporcionada.

```
<?php
include './header.html';
//Use connection.php to access the functions that are needed.
require_once 'connection.php';

//When the form is posted
if ($_SERVER["REQUEST_METHOD"] == "POST"){
    $userMail=$_POST['m'];

    /**This part is error checking. If there is no errors checkData() function is called in order to update the new password and a confi
    -The passwords don't match.
    -The data introduced is wrong.*/
    if(empty($_POST['m'])){
        header("Location: passwdRecovery.php?err2=true");
    }else{
        $mail = CheckMail($userMail);
        if($mail=="wrongmail"){
            header("Location: passwdRecovery.php?err=true");
        }else{
            header("Location: passwdRecovery.php?err=false");
        }
    }
}
}
?>
<!DOCTYPE html>
<!-- HTML password recovery form:
The form action is done in the same file using 'PHP_SELF'.
In this form it is introduced the user name, the verification values, the new password and the re-write of the new password.
There is also a link to go back to the login.
-->
<html>
<head>
    <title>Recovery</title>
    <meta charset = "UTF-8" name="viewport" content="width=device-width, initial-scale=1">
    <link href= "../css/recovery.css" rel="stylesheet" type="text/css">
    <link href= "../css/header-footer.css" rel="stylesheet" type="text/css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
    <script src= "../js/menuFunction.js"></script>
</head>
<body>
<section class="main">
    <article>
```

Figura 9.2.11: passwdRecovery.php

```
function checkMail($m){
    $users=connection();
    $count=0;
    foreach ($users as $row) {
        if($row['mail']==$m){
            $us=$row['name'];
            $pass=$row['passwd'];
            require "../composer/vendor/autoload.php"; //IMPORTANT
            $mail = new PHPMailer();
            $mail->IsSMTP();
            $mail->SMTPDebug = 0;
            $mail->SMTPAuth = true;
            $mail->SMTPSecure = "tls";
            $mail->Host = "smtp.gmail.com";
            $mail->Port = 587;
            $mail->Username = "IMPORTANT";
            $mail->Password = "IMPORTANT";
            $mail->SetFrom('IMPORTANT', 'Tasker');
            $mail->Subject = "Password recovery";
            $mail->MsgHTML('Hello '.$us." your password is ".$pass);
            $address = $m;
            $mail->AddAddress($address, "Tasker");
            $resul = $mail->Send();
            if(!$resul) {
                echo "Error" . $mail->ErrorInfo;
            } else {
                echo "Sent";
            }
            $count++;
        }
    }
    if($count<1){
        $ma="wrongmail";
    }else{
        $ma='';
    }
    return $ma;
}
```

Figura 9.2.12: checkMail() from connection.php

- createAccount.php

Este archivo contiene un formulario para crear un nuevo usuario. Los datos se envían al mismo archivo y se validan. Una vez hecho esto, si los datos son correctos se insertará un nuevo usuario en la base de datos y se enviará un correo de bienvenida al usuario. Esto se hace mediante la función newUser() de connection.php.

El email se envía igual que cuando se quiere recuperar la contraseña, es decir, usando PHPMailer, después se crea una query en formato String que en este caso sería un insert con los datos proporcionados en el formulario. Es interesante comentar que se usa la función password\_hash() de PHP al introducir la contraseña en la query, lo que se consigue con esto es que las contraseñas lleguen encriptadas a la base de datos.

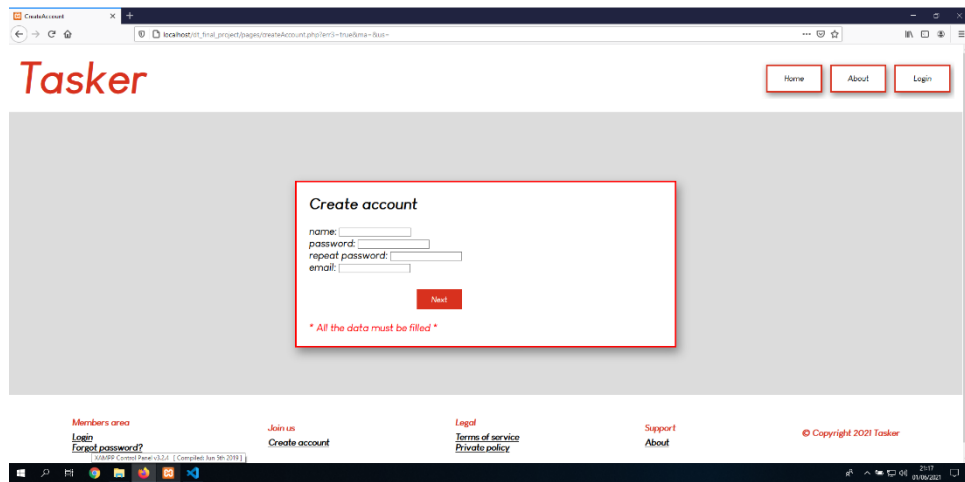


Figura 9.2.13: Vista de createAccount.php

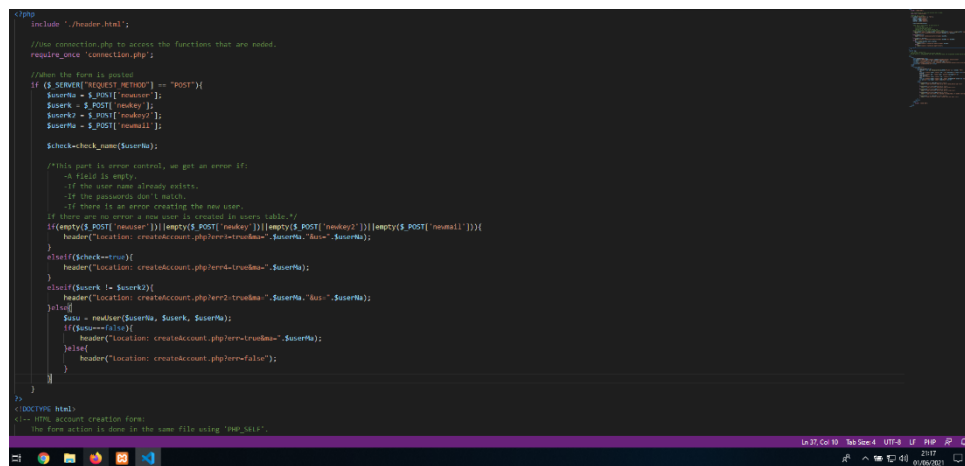


Figura 9.2.14: createAccount.php

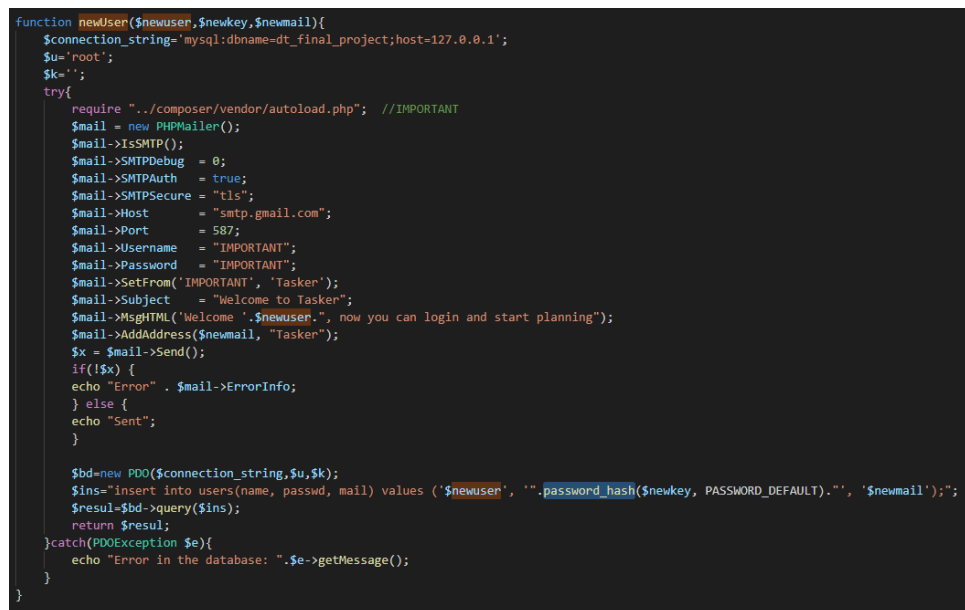


Figura 9.2.15: newUser() from connection.php

## 9.3 Aplicación

- Tasks

Cuando se inicia sesión entramos al main.php, un archivo que contiene el menú lateral desde el que nos movemos entre las diferentes secciones de la página, un título <h1> que da la bienvenida al usuario y un menú de usuario que veremos más adelante. Por defecto la sección que aparece es la crear tareas. En main.php se incluyen los archivos main.js y main.css que corresponden a las funciones y estilos de esta parte de la web.

En el centro de la página hay un botón para crear tareas y en el contenedor de la izquierda es donde se visualizan las tareas ya creadas. Las tareas creadas se visualizan gracias al archivo showTasks.php que está incluido en el código mediante un require\_once de php.

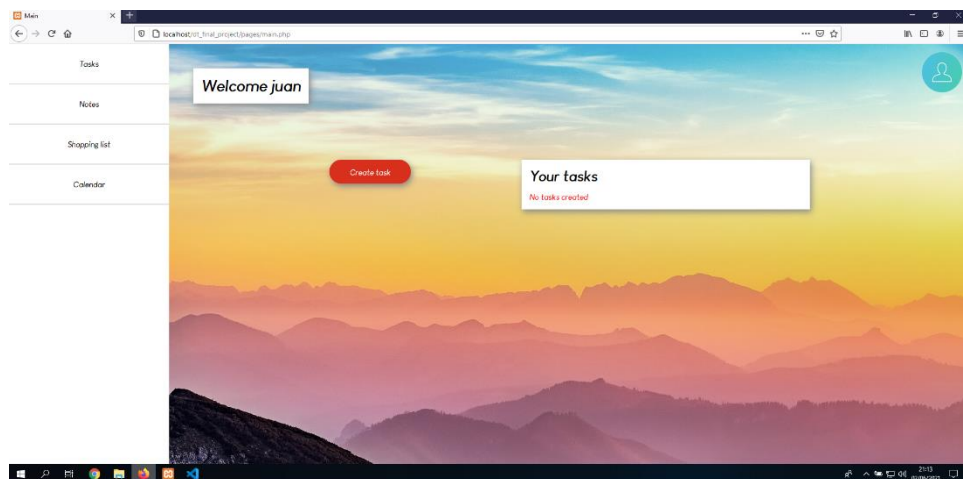


Figura 9.3.1: View of tasks from main.php



```

main.php x showTasks.php createTask.php JS main.js
C:\xampp\htdocs> cd final_project > pages > main.php
47 ?>
48 <!DOCTYPE html>
49 <!-- HTML main page:
50 If the login checking is correct we can access this file. This page is meant to be an index to access the application functions.
51 -->
52 <html>
53 <head>
54 <title>Main</title>
55 <meta charset = "UTF-8">
56 <link href="../css/main.css" rel="stylesheet" type="text/css">
57 <script src="../js/main.js"></script>
58 </head>
59 <body>
60 <div id="container">
61 <div id="menu">
62 <ul class="menu">
63 <li class="menu2">Tasks</li>
64 <li class="menu2">Notes</li>
65 <li class="menu2">Shopping list</li>
66 <li class="menu2">Calendar</li>
67 </ul>
68 </div>
69 <div id="content">
70 <?php
71 require("../header.php");
72 echo "h1>Welcome " .$_SESSION['user']. "</h1>";
73 >
74 <input id="hidden" type="hidden" value="<?php $u?>">
75 <div id="inside">
76 <div>Create task</div>
77 <div>
78 <h2>Your tasks</h2>
79 <?php require_once("../showTasks.php"); >
80 </div>
81 </div>
82 </div>
83 </body>
84 </html>
85

```

Figura 9.3.2: main.php

El archivo showTasks.php recoge las tareas creadas por un usuario mediante la función getTasks() de connection.php, si no hay tareas se mostrará un mensaje, pero si las hay se recorren los datos obtenidos y diferenciando por fecha (hoy, mañana u otra fecha mayor) se guardan en arrays a menos que la fecha esté atrasada, en este caso se elimina la tarea de la base de datos. Tras haber recogido las fechas en los arrays se muestran por pantalla las tareas diferenciando por fecha.

```

main.php x showTasks.php createTask.php JS main.js
C:\xampp\htdocs> cd final_project > pages > showTasks.php
15 $count=countTasks($cod);
16 if($count==0){
17 echo "<p style='color:red'>No tasks created</p>";
18 }else{
19 $today = [];
20 $tomorrow = [];
21 $dayday = [];
22 $dateow = date("Y-m-d");
23 $timeow = date("H:i:s");
24
25 $all=getTasks($cod);
26 foreach ($all as $row) {
27 $d = strtotime($row['date']);
28 $date = date("d/m/Y", $d);
29
30 $diff = abs($d - strtotime($dateow));
31 $years = floor($diff / (365*60*60*24));
32 $months = floor(($diff - $years * 365*60*60*24) / (30*60*60*24));
33 $days = floor(($diff - $years * 365*60*60*24 - $months*30*60*60*24) / (60*60*24));
34
35 if($row['date']==$dateow && $row['time']==$timeow){
36 array_push($today, $row['text'], $row['time'], $row['date']);
37 }elseif($row['date']==$dateow && $days==1){
38 array_push($tomorrow, $row['text'], $row['time'], $row['date']);
39 }elseif($row['date']==$dateow && $days>1){
40 array_push($dayday, $row['text'], $row['time'], $row['date']);
41 }elseif($row['date']==$dateow && $days>1){
42 deleteTask($row['text'], $row['date'], $row['time'], $cod);
43 }
44 }
45 echo "<div class='days'><h3>Today</h3>";
46 foreach($today as $day){
47 $arr = explode("-", $day);
48 echo "<div class='task'><span>".$arr[0]. " at ".substr($arr[1], 0, 5). "</span><span class='hiddenSpan'>".$arr[2]. "</span><span class='hiddenSpan'>".$arr[1]. "</span><div class='checkTask'></div></div>";
49 }
50 echo "</div>";
51 echo "<div class='days'><h3>Tomorrow</h3>";
52 foreach($tomorrow as $day){
53 $arr = explode("-", $day);
54 echo "<div class='task'><span>".$arr[0]. " at ".substr($arr[1], 0, 5). "</span><span class='hiddenSpan'>".$arr[2]. "</span><span class='hiddenSpan'>".$arr[1]. "</span><div class='checkTask'></div></div>";
55 }
56 echo "</div>";
57

```

Figura 9.3.3: showTasks.php

Cuando el usuario pulsa el botón Create Task en la aplicación se activa un evento que llama a la función create() de main.js. Esta función hace una llamada de

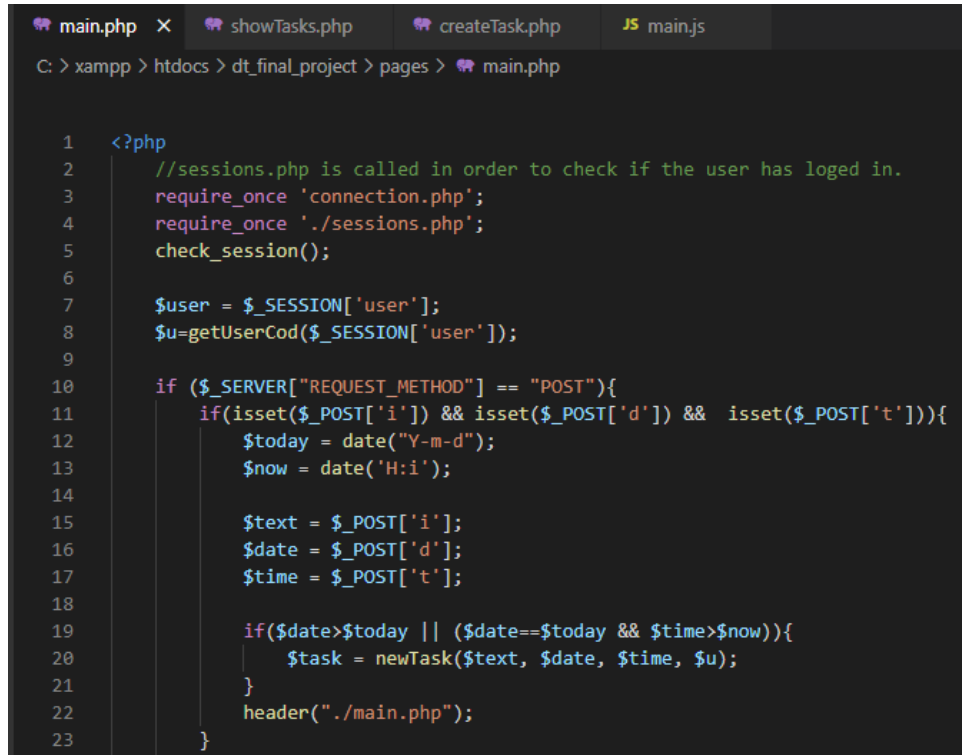
AJAX que recoge el contenido del archivo createTask.php y lo inserta en el contenedor de la derecha en main.php. Este contenido es un formulario con un input de texto, un input de fecha y un input de tiempo formateados ambos desde createTask.php. Este formulario se envía a main.php, desde donde se recogen los datos mediante el método POST y se llama a la función newTask de connection.php pasándole esta información no sin antes haberla validado. Por último, desde main.php se crea un header hacia este mismo archivo para recargar la página y que se muestre la nueva tarea.

```
function create(){
    document.querySelector("#inside > div:first-child").addEventListener("click", function(){
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
            if(this.readyState==4 && this.status==200) {
                document.querySelector("#inside > div:last-child").innerHTML=this.response;
                suggestions();
            }
        };
        xhttp.open("GET", "../pages/createTask.php", true);
        xhttp.send();
    });
}
```

Figura 9.3.4: método create() de main.js

```
main.php showTasks.php createTask.php X JS main.js
C:\xampp\htdocs> dt_final_project > pages > createTask.php
1 <?php
2 $today = date("Y-m-d");
3 $nextYear = (date("Y")+1)."-".date("m-d");
4
5 $nowH = date("H");
6 $nowH2 = date("H");
7 $nowM = date("i")+30;
8 $nowM2 = date("i")+5;
9 if($nowM==60){
10     $nowH++;
11     $nowM = str_pad(($nowM-60), 2, "0", STR_PAD_LEFT);
12     $nowH = str_pad($nowH, 2, "0", STR_PAD_LEFT);
13     if($nowM==24){
14         $nowH=0;
15         $nowM = str_pad($nowM, 2, "0", STR_PAD_LEFT);
16     }
17 }
18 }
19 <h2>Create task</h2>
20 <form action = "/main.php" method = "POST" autocomplete="off">
21     <div id="auto" class="suggestions">
22         <input type="text" id="t" name="t" maxlength="30" placeholder="I want to..." required>
23     </div><br>
24     On: <input type="date" id="d" name="d" value="<?php echo $today;>" min="<?php echo $today;>" max="<?php echo $nextYear;>" required><br>
25     At: <input type="time" id="t" name="t" value="<?php echo $nowH."<?php echo $nowM;>" required><br>
26     <input type="submit" value="Next">
27 </form>
```

Figura 9.3.5: createTask.php



```

1  <?php
2  //sessions.php is called in order to check if the user has logged in.
3  require_once 'connection.php';
4  require_once './sessions.php';
5  check_session();
6
7  $user = $_SESSION['user'];
8  $u=getUserCod($_SESSION['user']);
9
10 if ($_SERVER["REQUEST_METHOD"] == "POST"){
11     if(isset($_POST['i']) && isset($_POST['d']) && isset($_POST['t'])){
12         $today = date("Y-m-d");
13         $now = date('H:i');
14
15         $text = $_POST['i'];
16         $date = $_POST['d'];
17         $time = $_POST['t'];
18
19         if($date>$today || ($date==$today && $time>$now)){
20             $task = newTask($text, $date, $time, $u);
21         }
22         header("./main.php");
23     }

```

Figura 9.3.6: Recogida de datos de formulario en main.php

Una vez creada la tarea, esta se visualizará en el apartado de Your Tasks. Cada tarea tiene un icono de un check al lado que al ser clickado activa un evento de main.js que se encuentra dentro de la función deleteTask(). Este evento recoge los datos de esa tarea, es decir, el texto, la fecha y la hora y envía esos datos a showTasks.php mediante una llamada AJAX. Desde showTasks.php se llama a las funciones deleteTask y earn points de connection.php. La primera elimina la tarea en cuestión y la segunda añade 25 puntos al usuario.

Tras recibir la llamada, el evento llama a las funciones updatePoints() y create() de main.js. La segunda función ya la he comentado anteriormente, pero updatePoints() realiza una llamada de AJAX a updatePoints.php que devuelve los puntos que tiene el usuario, esta respuesta sirve para actualizar el apartado points del menú de usuario que se comenta en el siguiente apartado de esta sección.

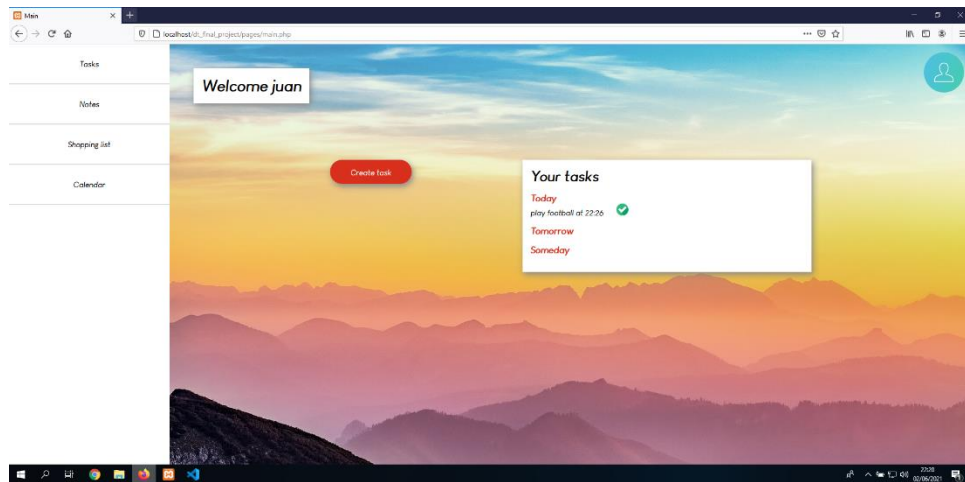


Figura 9.3.7: Vista de tareas creadas desde main.php

```
function deleteTask() {
    var ts = document.querySelectorAll("#inside > div:last-child > div > div > .checkTask");
    if(!ts) return;
    for (var i=0; i<ts.length; i++) {
        ts[i].addEventListener("click", function () {
            var str = this.parentNode.querySelector("p").innerHTML;
            var d = this.parentNode.querySelector(".hiddenSpan1").innerHTML;
            var h = this.parentNode.querySelector(".hiddenSpan2").innerHTML;
            var xhttp = new XMLHttpRequest();
            xhttp.onreadystatechange = function () {
                if (this.readyState == 4 && this.status == 200) {
                    document.querySelector("#inside").innerHTML = "<div>Create task</div><div><h2>Your tasks</h2>" + this.response + "</div>";
                    updatePoints();
                    create();
                }
            };
            xhttp.open("GET", "../pages/showTasks.php?text=" + str.substr(0, str.indexOf(' at ')) + "&date=" + d + "&time=" + h, true);
            xhttp.send();
        });
    }
}
```

Figura 9.3.8: Método con el eventListener del check de tarea.

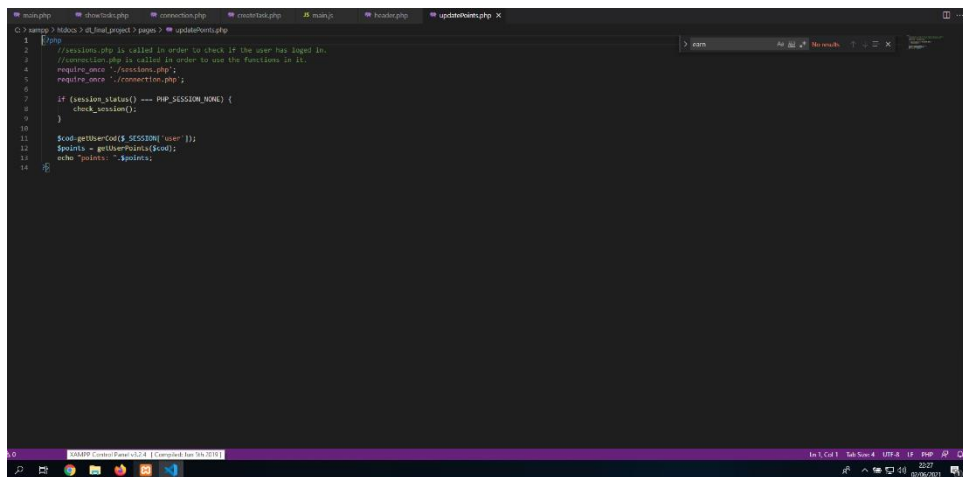
```
main.php showTasks.php X connection.php createTask.php JS main.js head...
C: > xampp > htdocs > dt_final_project > pages > showTasks.php

1  <?php
2  require_once './connection.php';
3  require_once './sessions.php';
4  if (session_status() === PHP_SESSION_NONE) {
5      check_session();
6  }
7
8  $cod=getUserCod($_SESSION['user']);
9
10 if(isset($_GET['text']) && isset($_GET['date']) && isset($_GET['time'])){
11     deleteTask($_GET["text"], $_GET["date"], $_GET["time"], $cod);
12     earnPoints($cod);
13 }
14
```

Figura 9.3.9: Recogida de datos en showTasks.php

```
function updatePoints(){
    var xhttp = new XMLHttpRequest();
    xhttp.onreadystatechange = function () {
        if (this.readyState == 4 && this.status == 200) {
            document.querySelector("#myDropdown > a:nth-child(2)").innerHTML = this.response;
            //updatePoints();
            create();
            deleteTask();
        }
    };
    xhttp.open("GET", "../pages/updatePoints.php", true);
    xhttp.send();
}
```

Figura 9.3.10: Función updatePoints() de main.js



```
1 //updatePoints.php
2 //sessions.php is called in order to check if the user has logged in.
3 //connection.php is called in order to use the functions in it.
4 require_once "../sessions.php";
5 require_once "../connection.php";
6
7 if (session_status() == PHP_SESSION_NONE) {
8     check_session();
9 }
10
11 $code = getHeaderCode($SESSION['user']);
12 $points = getHeaderPoints($code);
13
14 echo "points: " . $points;
```

Figura 9.3.11: updatePoints.php

- Header.php

Este archivo contiene un menú desplegable con el nombre de usuario, los puntos que tiene el usuario por haber completado tareas y un enlace al archivo logout.php para cerrar sesión y cuyo funcionamiento se explicó en la sección 9.2.

El menú se despliega mediante un evento de click en main.js y haciendo uso de las clases correspondientes para visualizarlo o no. Estas clases se encuentran en main.css.

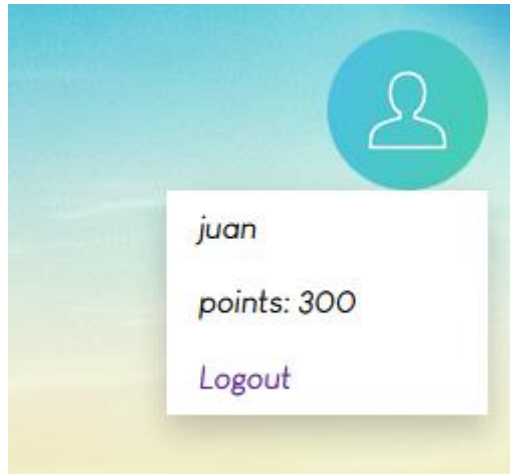


Figura 9.3.12: Vista del menú desplegable desde main

```
File Edit Selection View Go Run Terminal Help
main.php showTasks.php connection.php createTask.php JS main.js
C: > xampp > htdocs > dt_final_project > pages > header.php
1 <?php
2 //sessions.php is called in order to check if the user has logged in.
3 //connection.php is called in order to use the functions in it.
4 require_once './sessions.php';
5 require_once './connection.php';
6
7
8 <div class="dropdown">
9   <div id="dropbtn" class="dropbtn"></div>
10  <nav id="myDropdown" class="dropdown-content">
11    <a><?php
12      echo $_SESSION['user'];
13    ?></a>
14    <a><?php
15      $cod=getUserCod($_SESSION['user']);
16      $points = getUserPoints($cod);
17      echo "points: ".$points;
18    ?></a>
19    <a href="./logout.php">Logout</a>
20  </nav>
21 </div>
```

Figura 9.3.13: header.php

```
//Display or not the dropdown when clicking the user's icon at main page
document.getElementById("dropbtn").addEventListener("click", myFunction);
function myFunction() {
    document.getElementById("myDropdown").classList.toggle("show");
}
window.onclick = function(event) {
    if (!event.target.matches('.dropbtn')) {
        var dropdowns = document.getElementsByClassName("dropdown-content");
        var i;
        for (i = 0; i < dropdowns.length; i++) {
            var openDropdown = dropdowns[i];
            if (openDropdown.classList.contains('show')) {
                openDropdown.classList.remove('show');
            }
        }
    }
}
```

Figura 9.3.14: Evento para desplegar el menú.

- Notes

Si clicamos en la sección Notes del menú lateral izquierdo en el contenido de la página se mostrará un botón para crear notas y se carga la respuesta del archivo showNotes.php en el contenedor de la derecha en el centro de la página. El archivo showNotes.php muestra las notas creadas por el usuario, para ello llama al método getNotes() de connection.php que devuelve todas las notas que ha creado un usuario, este array se recorre y se crean contenedores para cada nota.

Si se pulsa el botón Create note se activa un evento de main.js que carga en el contenedor de la derecha de main.php el contenido de createNote.php, es decir, un formulario con un solo input de texto que se enviará a main.php. Desde main.php se llama a la función newNote() de connection.php que inserta la nueva nota en la base de datos. También se hace un echo del script notes.js el cual muestra la lista con las notas creadas.

Las notas creadas pueden ser eliminadas al clicar en el icono de la papelera, esto se hace con un evento que mediante una función de AJAX carga el contenido de showTasks.php pero además pasa a este archivo el texto de la nota a eliminar, este texto se pasa a la función deleteNote() de connection.php que elimina la nota de la base de datos.

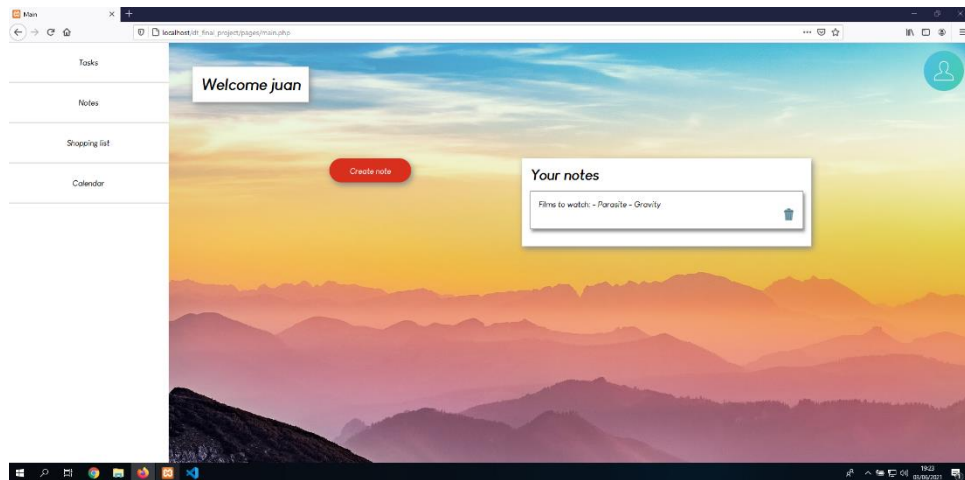


Figura 9.3.15: Vista de la sección Notes en main.php.

```

main.php JS notes.js connection.php JS main.js createNote.php showNotes.php X
C: > xampp > htdocs > dt_final_project > pages > showNotes.php
1  <?php
2      require_once './connection.php';
3      require_once './sessions.php';
4      if (session_status() === PHP_SESSION_NONE) {
5          check_session();
6      }
7
8      $cod=getUserCod($_SESSION['user']);
9      if(isset($_GET['text'])){
10         deleteNote($cod, $_GET["text"]);
11     }
12     $count=countNotes($cod);
13     if($count==0){
14         echo "<p style='color:red'>No notes created</p>";
15     }else{
16         $all=getNotes($cod);
17         foreach ($all as $row) {
18             echo "<div class='n' contentEditable='true'><p>".$row['text'].</p><div class='trash'></div></div><br>";
19         }
20     }

```

Figura 9.3.16: showNotes.php

```

main.php JS notes.js connection.php JS main.js createNote.php X showNotes.php
C: > xampp > htdocs > dt_final_project > pages > createNote.php
1  <h2>Create Note</h2>
2  <form action = "../main.php" method = "POST">
3  <textarea id="note" name="note" rows="5" cols="30" placeholder="Write here" required></textarea>
4  <br>
5  <input type = "submit" value="Next">
6  </form>

```

Figura 9.3.17: createNote.php



```
function newNote($text, $user){
    $connection_string='mysql:dbname=dt_final_project;host=127.0.0.1';
    $u='root';
    $k='';
    try{
        $bd=new PDO($connection_string,$u,$k);
        $ins="insert into notes(text, codUser) values ('$text','$user')";
        $resul=$bd->query($ins);
        return $resul;
    }catch(PDOException $e){
        echo "Error in the database: ".$e->getMessage();
    }
}
```

Figura 9.3.18: Método newNote de connection.php

```
main.php JS notes.js X connection.php JS main.js createNote.php showNotes.php
C:\xampp\htdocs> dt_final_project > js > notes.js > note
1 function note() {
2     var xhttp = new XMLHttpRequest();
3     xhttp.onreadystatechange = function() {
4         if(this.readyState==4 && this.status==200) {
5             document.querySelector("#inside").innerHTML="<div>Create note</div><div><h2>Your notes</h2>"+this.response+"</div>";
6             createNote();
7         }
8     };
9     xhttp.open("GET", "../showNotes.php", true);
10    xhttp.send();
11 }
12 function createNote() {
13     var not = document.querySelectorAll("#inside > div:last-child > .n > div");
14     for(var i=0;i<not.length;i++){
15         not[i].addEventListener("click", function(){
16             var xhttp = new XMLHttpRequest();
17             xhttp.onreadystatechange = function() {
18                 if(this.readyState==4 && this.status==200) {
19                     document.querySelector("#inside").innerHTML="<div>Create note</div><div><h2>Your notes</h2>"+this.response+"</div>";
20                     createNote();
21                 }
22             };
23             xhttp.open("../showNotes.php?text="+this.parentNode.querySelector("p").innerHTML, true);
24             xhttp.send();
25         });
26     }
27 }
28 document.querySelector("#inside > div:first-child").addEventListener("click", function(){
29     var xhttp = new XMLHttpRequest();
30     xhttp.onreadystatechange = function() {
31         if(this.readyState==4 && this.status==200) {
32             document.querySelector("#inside > div:last-child").innerHTML=this.response;
33         }
34     };
35     xhttp.open("GET", "../createNote.php", true);
36     xhttp.send();
37 });
38 }
39 note();
```

Figura 9.3.19: notes.js

- Shopping list

Al clicar en la sección Shopping list del menú lateral de la izquierda se mostrará un botón Create list y el contenido del archivo showList.php, desde este archivo se llama a la función getList() de connection.php que recoge todos los productos que el usuario ha añadido a su lista de la compra. Se recorre el array obtenido diferenciando entre ellos en función de su texto para poner una imagen u otra en función del producto.

Los productos pueden ser eliminados de la lista al clicar sobre ellos o en el check verde, esto se hace mediante un evento dentro de la función createShop() de main.js en la línea 191 que hace una llamada de AJAX a deleteProduct.php, desde este archivo se llama al método deleteProduct() de connection.php que elimina el producto tachado.

Si el usuario pulsa el botón Create list se cargará en la página una lista de productos que al clicarlos se añadirán a la lista de la compra, esto se hace en la línea 291, se clona el producto clicado y se añade a otro contenedor que se encuentra debajo. A este producto clonado se le añade un evento para poder ser tachado o no en el momento (antes de crear la lista), esto se hace desde el mismo evento donde se clona. En un formulario oculto se añaden inputs con los nombres los productos añadidos a la lista en un array y las cantidades en otro array. Estos datos son enviados a main.php cuando el usuario clicca el botón Next.

Además de seleccionar productos predefinidos el usuario puede añadir sus propios productos al rellenar el input de arriba y clicar el icono a su izquierda. Estos productos tienen una imagen diferente a los productos predefinidos. Esta funcionalidad se consigue a partir de la línea 206 en un evento de click para el icono mencionado anteriormente que recoge los datos del input, crea un div de producto nuevo añadiéndole la imagen del carrito, los eventos para tacharlo y guardando el texto del producto y la cantidad en los arrays correspondientes.

La cantidad de los productos se cambia haciendo uso de un evento onchange para el input en cuestión.

Una vez añadidos todos los productos desados a la lista, al clicar en el botón Next los datos son enviados a main.php, desde donde se llama a la función newProduct() de connection.php a la que se le pasa como parámetros los datos de los arrays de productos y cantidades de los mismos tras haber sido recorridos. La función newProduct inserta el nuevo producto en base de datos. Finalmente desde main.php se hace un echo del script shoppingList.js con el objetivo de visualizar la lista de la compra con los nuevos productos y volver a añadir funcionalidad a los elementos.

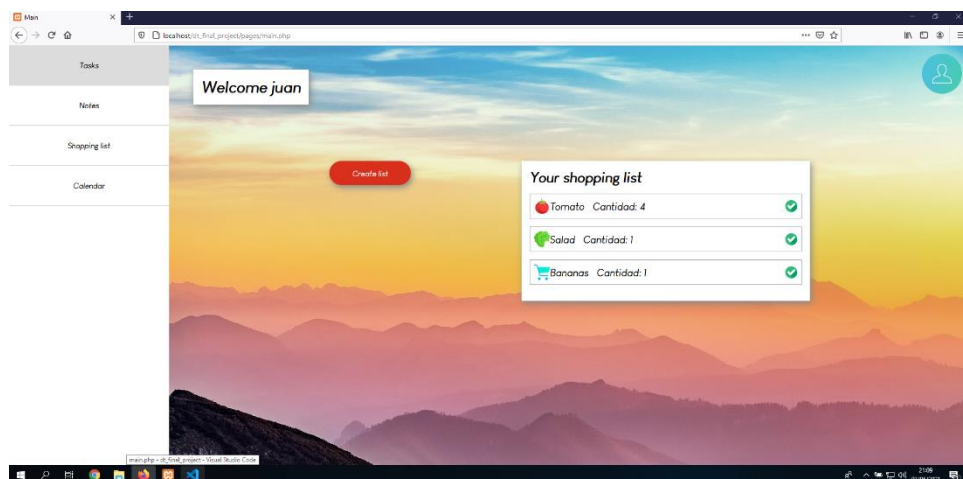
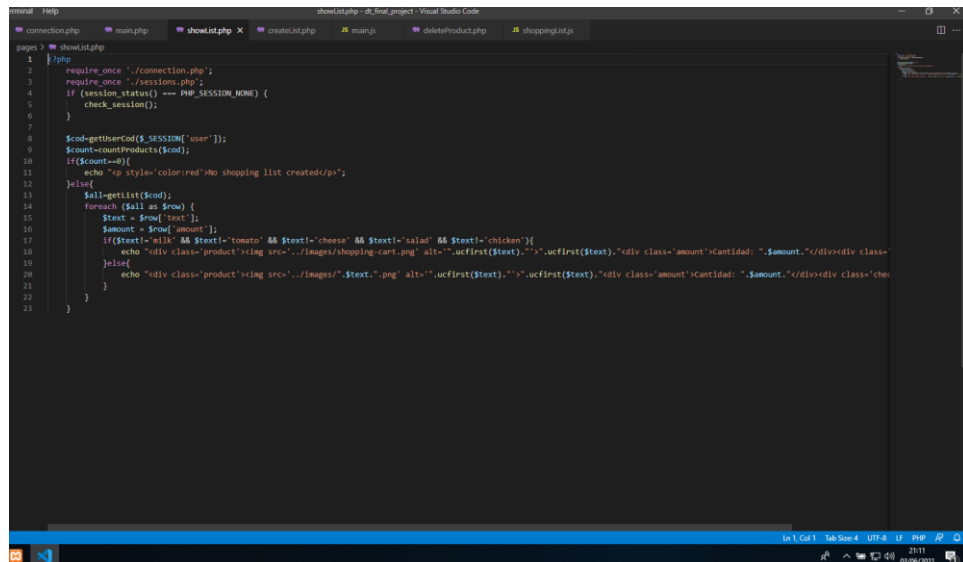


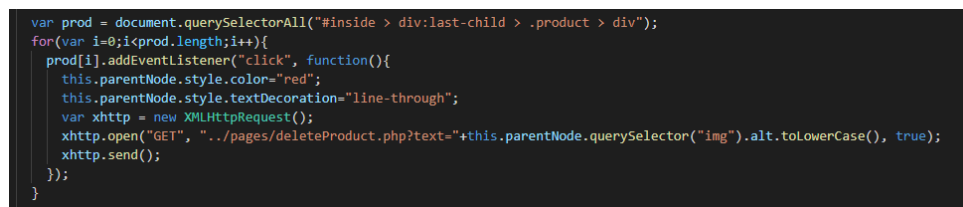
Figura 9.3.20: Vista de la lista de la compra desde main.php



```

1  <?php
2  require_once './connection.php';
3  require_once './sessions.php';
4  if (session_status() === PHP_SESSION_NONE) {
5      check_session();
6  }
7
8  $cod=getUserCod($_SESSION['user']);
9  $count=countProducts($cod);
10 if($count==0){
11     echo "<p style='color:red'>No shopping list created</p>";
12 }else{
13     $all=getList($cod);
14     foreach ($all as $row) {
15         $text = $row['text'];
16         $amount = $row['amount'];
17         if($text!='milk' && $text!='tomato' && $text!='cheese' && $text!='salad' && $text!='chicken'){
18             echo "<div class='product'><img src='../images/shopping-cart.png' alt='".$ucfirst($text)."'></div><div class='
19         }else{
20             echo "<div class='product'><img src='../images/".$text.".png' alt='".$ucfirst($text)."'></div><div class='che
21         }
22     }
23 }
    
```

Figura 9.3.21: showList.php



```

var prod = document.querySelectorAll("#inside > div:last-child > .product > div");
for(var i=0;i<prod.length;i++){
    prod[i].addEventListener("click", function(){
        this.parentNode.style.color="red";
        this.parentNode.style.textDecoration="line-through";
        var xhttp = new XMLHttpRequest();
        xhttp.open("GET", "../pages/deleteProduct.php?text="+this.parentNode.querySelector("img").alt.toLowerCase(), true);
        xhttp.send();
    });
}
    
```

Figura 9.3.22: Evento para eliminar productos de la lista de la compra en main.js

```

js > JS main.js > document.addEventListener("DOMContentLoaded") callback > createShop > addEventListen
285      document.getElementById( product ).value= ;
286    });
287
288    var products = document.querySelectorAll("#products > div");
289    for(var i=0;i<products.length;i++){
290
291      products[i].addEventListener("click", function(){
292        if(!this.classList.contains("clicked")) {
293          var clone = this.cloneNode(true);
294
295          clone.addEventListener("click", function(){
296            if(this.classList.contains("crossed")){
297              var val = (this.querySelector("img").alt).toLowerCase();
298              var a = document.querySelectorAll("#items > input");
299              for(var i=0;i<a.length;i++){
300                if(a[i].value==val){
301                  a[i].name="array[]";
302                }
303              }
304              this.classList.add("notCrossed");
305              this.classList.remove("crossed");
306            }else{
307              var val2 = (this.querySelector("img").alt).toLowerCase();
308              var a2 = document.querySelectorAll("#items > input");
309              for(var i=0;i<a2.length;i++){
310                if(a2[i].value==val2){
311                  a2[i].name="";
312                }
313              }
314              this.classList.add("crossed");
315              this.classList.remove("notCrossed");
316            }
317          });
318          var inp2 = document.createElement("input");
319          inp2.type="hidden";
320          inp2.name="array[]";
321          inp2.value=(this.querySelector("img").alt).toLowerCase();
322          document.getElementById("items").appendChild(inp2);
323
324          var contProduct = document.createElement("div");
325          contProduct.classList.add("contProduct");
326          contProduct.appendChild(clone);
327

```

Figura 9.3.23: Evento para añadir productos a la lista de la compra y funcionalidad a estos en main.js

```

205
206     document.getElementById("add").addEventListener("click", function(){
207         var text = document.getElementById("product").value;
208         if(!text.length==0) {
209             var ele = document.createElement("div");
210             ele.classList.add("product");
211             var img = document.createElement("img");
212             img.src = "../images/shopping-cart.png";
213             img.alt = text.toLowerCase();
214             ele.appendChild(img);
215             var t = document.createElement("span");
216             t.innerHTML=text;
217             ele.appendChild(t);
218
219             var inp = document.createElement("input");
220             inp.type="hidden";
221             inp.name="array[]";
222             inp.value=text.toLowerCase();
223             document.getElementById("items").appendChild(inp);
224
225             ele.addEventListener("click", function(){
226                 if(ele.classList.contains("crossed")){
227                     var val = (this.querySelector("img").alt).toLowerCase();
228                     var a = document.querySelectorAll("#items > input");
229                     for(var i=0;i<a.length;i++){
230                         if(a[i].value==val){
231                             a[i].name="array[]";
232                         }
233                     }
234                     ele.classList.add("notCrossed");
235                     ele.classList.remove("crossed");
236                 }else{
237                     var val2 = (this.querySelector("img").alt).toLowerCase();
238                     var a2 = document.querySelectorAll("#items > input");
239                     for(var i=0;i<a2.length;i++){
240                         if(a2[i].value==val2){

```

Figura 9.3.24: Evento para añadir productos personalizados a la lista en main.js

```

if(isset($_POST['array'])){
    $stringAmounts=$_POST['arrayCant'];
    $arrayAmounts = explode(",", $stringAmounts);
    $arrayProducts=$_POST['array'];

    if(countProducts($u)>0){
        deleteProducts($u);
    }

    $length = count($arrayProducts);
    for($i=0; $i<$length; $i++){
        newProduct($arrayProducts[$i], $u, $arrayAmounts[$i]);
    }
    echo '<script src="../js/shoppingList.js"></script>';
}

```

Figura 9.3.25: Recogida de los arrays con productos en main.php

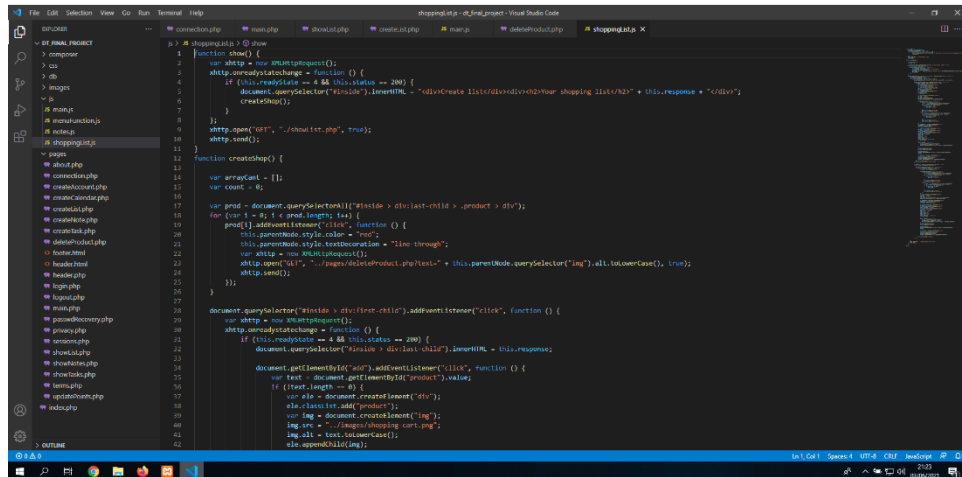


Figura 9.3.26: shoppingList.js

- Calendar

Al clicar en la sección Calendar de la página el archivo createCalendar.php es cargado mediante una llamada AJAX en el contenedor principal de la página.

En createCalendar.php hay una función llamada build\_calendar() que es para crear el calendario. En esta función se crea un array con los días de la semana, se obtiene el primer día del mes, el número de días del mes, nombre del mes. El calendario se crea concatenando los elementos de tabla en una cadena de String para luego hacer un echo y mostrarlos por pantalla.

Se recorre el array con los días de la semana para crear las cabezeras de las columnas de la tabla. Se hace un bucle usando el número de días del mes y cuando ese bucle llega a 7 la siguiente columna será la 0. Mientras se está dentro de este bucle se comprueba si para esa fecha hay alguna tarea programada usando el método getTasks() de connection.php, si la hay la celda que contiene al número del mes tendrá un color de letra rojo y un párrafo con la hora y texto de la tarea que se visualizará gracias a clases de main.css al pasar el ratón por encima del día.

Antes de crear el calendario también se añaden dos botones para cambiar de mes. Al hacer click en los botones se cambia el valor de la cookie calendar que recoge el año y mes del calendario. Al clicar los botones también se llama al archivo createCalendar.php pero al existir la cookie calendar al crear el calendario se usan el mes y el año almacenados en la cookie. El objetivo de esto es que el usuario pueda desplazarse entre los meses y años y que al cambiar de sección se mantenga el mes que estaba mirando por última vez.

La cookie calendar se elimina cuando el usuario cierra sesión, esto es en el archivo logout.php.

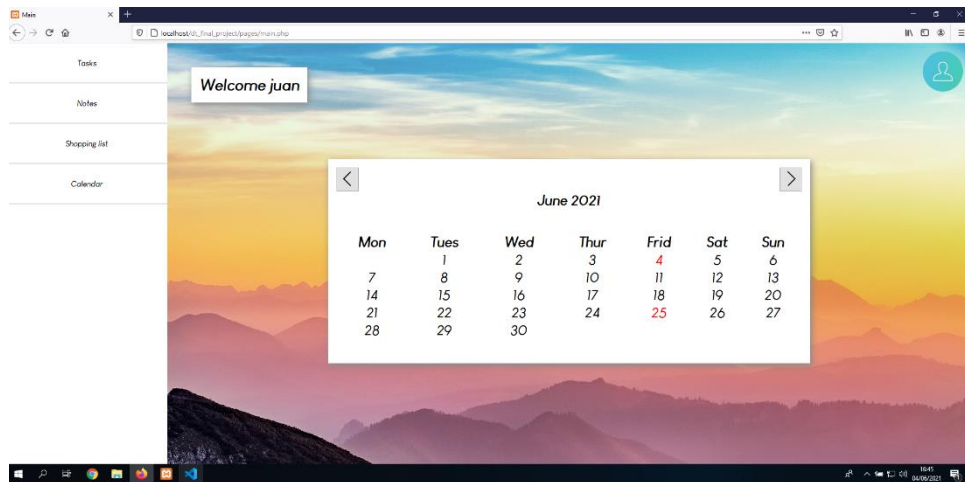


Figura 9.3.27: Vista del calendario desde main.

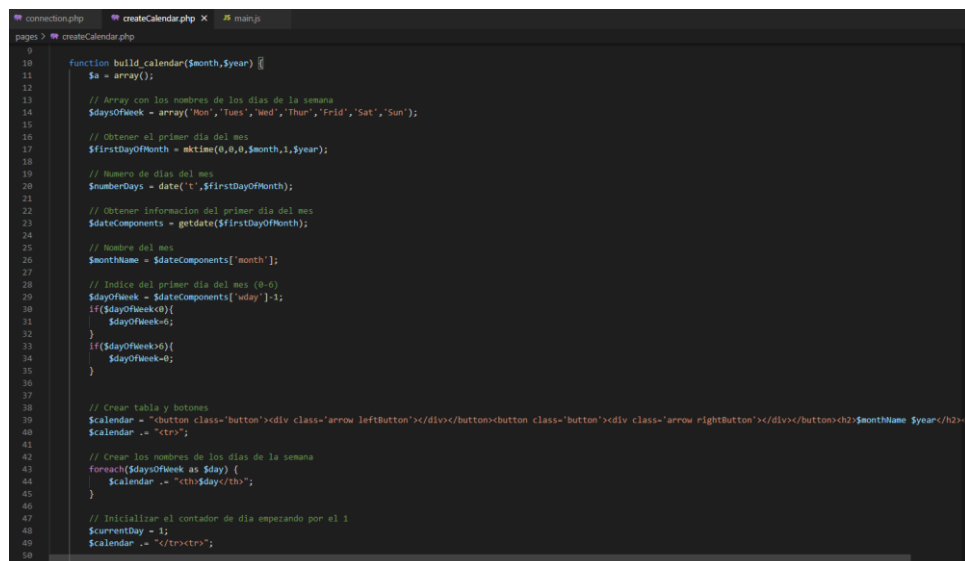


Figura 9.3.28: createCalendar.php 1ª parte

```

connection.php  createCalendar.php X  JS main.js
pages > createCalendar.php
78     }
79     if(strlen($text2) > $long) {
80         $text2 .="</div></div></td>";
81         $calendar .= $text2;
82     }else {
83         $calendar .= $text1;
84     }
85
86     // Incrementar contadores
87     $currentDay++;
88     $dayOfWeek++;
89 }
90
91 // Completar la lista de la ultima semana si es necesario
92 if ($dayOfWeek != 7) {
93     $remainingDays = 7 - $dayOfWeek;
94     $calendar .= "<td colspan='$remainingDays'>&nbsp;</td>";
95 }
96
97 $calendar .= "</tr>";
98 $calendar .= "</table>";
99
100 return $calendar;
101 }
102
103 $dateComponents = getdate();
104 $month = $dateComponents['mon'];
105 $year = $dateComponents['year'];
106
107 if(isset($_COOKIE['calendar'])){
108     $array = explode(",", $_COOKIE["calendar"]);
109     echo build_calendar($array[0], $array[1]);
110 }else{
111     echo build_calendar($month, $year);
112     setcookie("calendar",$month.", ".$year, time()+3600*24);
113 }
114 ?>
115 </div>

```

Figura 9.3.29: createCalendar.php 2ª parte



```

connection.php  createCalendar.php X  JS main.js
pages > createCalendar.php
//
78     }
79     if(strlen($text2) > $long) {
80         $text2 .="</div></div></td>";
81         $calendar .= $text2;
82     }else {
83         $calendar .= $text1;
84     }
85
86     // Incrementar contadores
87     $currentDay++;
88     $dayOfWeek++;
89 }
90
91 // Completar la lista de la ultima semana si es necesario
92 if ($dayOfWeek != 7) {
93     $remainingDays = 7 - $dayOfWeek;
94     $calendar .= "<td colspan='$remainingDays'>&nbsp;</td>";
95 }
96
97 $calendar .= "</tr>";
98 $calendar .= "</table>";
99
100 return $calendar;
101
102
103 $dateComponents = getdate();
104 $month = $dateComponents['mon'];
105 $year = $dateComponents['year'];
106
107 if(isset($_COOKIE['calendar'])){
108     $array = explode(",", $_COOKIE["calendar"]);
109     echo build_calendar($array[0], $array[1]);
110 }else{
111     echo build_calendar($month, $year);
112     setcookie("calendar",$month.", ".$year, time()+3600*24);
113 }
114 ?>
115 </div>

```

Figura 9.3.30: Eventos para los botones del calendario en main.js

## 10. Manual de usuario

Antes de explicar el manual de usuario conviene recordar que para poder usar la aplicación desde un entorno local hay que instalar XAMPP y realizar los pasos del punto 7.1 para situar el proyecto en nuestro equipo. Para poder usar los servicios de PHPMailer será necesario seguir los pasos indicados en el apartado 7.10 de esta memoria.

### Inicio

Nada más introducir la dirección del proyecto en el navegador, el usuario verá la página de inicio. Desde esta página podrá navegar a diferentes secciones desde el footer o el header.

Para iniciar sesión el usuario deberá introducir en la sección de “Login” sus credenciales. Si es la primera vez que entra en la página el usuario debería crearse

una nueva cuenta, para ello deberá dirigirse a la sección “Create account” desde el login o desde el footer.

En “Create account”, el usuario deberá introducir sus datos y una vez estén validados se enviará un correo a la dirección proporcionada. En este punto el usuario ya puede iniciar sesión con su nueva cuenta.

Si el usuario ha olvidado su contraseña deberá ir a la sección “Forgot password?”, desde donde recibirá un email con sus credenciales.

## Aplicación

Una vez el usuario ha iniciado sesión verá la sección de tareas de la aplicación. Para crear una tarea hay que clicar en el botón “Create task”. Las tareas se eliminan automáticamente si la fecha y horas indicadas han expirado. Por el contrario, si el usuario marca el check verde al lado de la tarea, esta desaparecerá y se considerará finalizada, recibiendo así el usuario 25 puntos por completarla.

El usuario puede ver sus puntos en el menú de usuario en la esquina superior derecha, desde aquí también se puede cerrar sesión.

En el menú lateral situado a la izquierda se puede navegar entre las diferentes herramientas que ofrece la aplicación. Aunque para móviles este menú se encuentra en la parte superior.

Para ver las notas hay que clicar en la sección correspondiente del menú lateral. Las notas se pueden crear desde el botón “Create note” y una vez creadas pueden eliminarse al pinchar en el icono de la papelera.

En la sección de lista de la compra se puede crear una lista o visualizar la lista actual. Si el usuario pulsa el botón para crear una, verá una serie de productos de muestra que al pulsarlos se añaden a una lista en la parte de abajo, estos productos añadidos pueden tacharse al clicar sobre ellos y también se puede cambiar la cantidad. Si el usuario desea añadir sus propios productos deberá introducir el nombre en el cuadro de texto de la parte superior y añadirlo pulsando el icono de añadir a la izquierda.

Una vez creada la lista de la compra los productos pueden eliminarse pulsando encima de ellos o en el check verde.

La última herramienta es el calendario, desde esta sección el usuario puede ir cambiando entre los diferentes meses y además podrá visualizar las tareas programadas para un día concreto, estos días están marcados en rojo y al pasar el ratón por encima se muestran las tareas programadas.

## 11. Conclusión

Tal y como se explicaba en el apartado de introducción y de objetivos, el principal era crear una aplicación web, que tuviera un diseño atractivo, fácil de entender y de utilizar por todos usuarios que se conecten mediante ordenador o dispositivo móvil.

Creo que he cumplido los objetivos que me había marcado ya que he realizado un proyecto donde tenemos una aplicación que permite organizar y gestionar el tiempo, así como elaborar tareas, lista de la compra accesible, etc. La he puesto a prueba entre mi círculo más cercano y les ha parecido útil, aunque me han sugerido mejoras como sería que te enviara una alerta o aviso visual y sonoro al móvil u ordenador de que está a punto de expirar el tiempo de inicio de una tarea programada.

También se puede mejorar consiguiendo puntos para obtener premios a través de la aplicación y crear ligas o concursos entre los usuarios o grupos de amigos y familiares, como incentivador en el uso y realización de las tareas programadas.

A nivel personal este proyecto me ha aportado una experiencia positiva y me siento satisfecho del resultado final. Elaborar un proyecto entero de inicio a fin es una prueba difícil, pero nos permite poner en práctica todo lo aprendido y superar este reto supone un gran esfuerzo, pero también una gran satisfacción a medida que se van solucionando las dificultades y se van obteniendo resultados.

He pretendido aplicar todos los conocimientos, así como las técnicas y métodos aprendidos a lo largo de estos dos años de formación.

La redacción de este documento me ha supuesto gran esfuerzo, ya que era la primera vez que tenía que realizar un proyecto y explicar de manera clara todos los pasos ejecutados para llegar a su fin, pero he intentado elaborarlo con la máxima calidad posible, aunque redactar un documento tan extenso es muy complicado.

Con todo lo anteriormente expuesto creo que he desarrollado un proyecto interesante, con muchas posibilidades y que puede ser ampliado en el futuro.

## 12. Referencias

- Bibliografía consultada para la realización de esta memoria:

<http://www.rhhdigital.com/secciones/actualidad/135934/La-importancia-de-las-agendas-para-aumentar-la-productividad>

<https://psicologiaymente.com/psicologia/tipos-de-motivacion>

<https://carlosmalfatti.com/disenio-web-equilibrio-entre-estetica-y-usabilidad/>

<https://northdigital3.com/la-importancia-del-diseno-la-estetica-pagina-web-una-presencia-online/>

<https://www.actualidadiphone.com/la-importancia-del-buen-diseno-de-una-aplicacion>

<http://shandweb.com/shandweb/article/la-importancia-de-un-buen-diseo-en-tu-app/71>

<https://diseno127244936.wordpress.com/2017/12/01/19/>

<https://galernaestudio.com/tipografias-web-tipos-de-fuentes/>

‘Lenguaje de Programación’. Autor: María Estela Raffino. De: Argentina.

Para: Concepto de Lenguaje de Programación. Disponible en:

<https://concepto.de/lenguaje-de-programacion/>. Última edición: 11 de julio de

2020

<https://www.muyinteresante.es/tecnologia/articulo/cuanto-tipos-de-lenguaje-de-programacion-existen-511584540297>

<https://www.marketingandweb.es/marketing/lenguajes-de-programacion-mas-usados/>

<https://www.universia.net/es/actualidad/orientacion-academica/5-herramientas-programacion-que-seran-tendencia-2019-1164773.html>

<https://ubiquim.com/es/blog/20-herramientas-de-desarrollo-de-software-que-te-haran-mas-productivo/>

- Imágenes empleadas en la página web:

<https://unsplash.com/photos/mfB1B1s4sMc>

<https://unsplash.com/photos/g1Kr4Ozfoac>

<https://pixabay.com/es/vectors/usuario-icono-persona-personal-2517433/>

<https://pixabay.com/es/photos/monta%C3%B1as-paisaje-silueta-3605113/>

[https://www.flaticon.com/free-icon/plus\\_1828925?term=plus&page=1&position=3&page=1&position=3&related\\_id=1828925&origin=search](https://www.flaticon.com/free-icon/plus_1828925?term=plus&page=1&position=3&page=1&position=3&related_id=1828925&origin=search)

[https://www.flaticon.com/free-icon/shopping-cart\\_833486?term=shop&related\\_id=833486](https://www.flaticon.com/free-icon/shopping-cart_833486?term=shop&related_id=833486)

[https://www.flaticon.com/free-icon/milk\\_684631?term=milk&page=1&position=1&page=1&position=1&related\\_id=684631&origin=search](https://www.flaticon.com/free-icon/milk_684631?term=milk&page=1&position=1&page=1&position=1&related_id=684631&origin=search)

[https://www.flaticon.com/free-icon/tomato\\_2909922?term=tomato&page=1&position=2&page=1&position=2&related\\_id=2909922&origin=search](https://www.flaticon.com/free-icon/tomato_2909922?term=tomato&page=1&position=2&page=1&position=2&related_id=2909922&origin=search)

[https://www.flaticon.com/free-icon/cheese\\_135651?term=cheese&page=1&position=8&page=1&position=8&related\\_id=135651&origin=search](https://www.flaticon.com/free-icon/cheese_135651?term=cheese&page=1&position=8&page=1&position=8&related_id=135651&origin=search)

[https://www.flaticon.com/free-icon/chicken-leg\\_1046802?term=chicken&page=1&position=12&page=1&position=12&related\\_id=1046802&origin=search](https://www.flaticon.com/free-icon/chicken-leg_1046802?term=chicken&page=1&position=12&page=1&position=12&related_id=1046802&origin=search)

[https://www.flaticon.com/free-icon/salad\\_1155281?term=salad&page=1&position=20&page=1&position=20&related\\_id=1155281&origin=search](https://www.flaticon.com/free-icon/salad_1155281?term=salad&page=1&position=20&page=1&position=20&related_id=1155281&origin=search)

[https://www.flaticon.es/icono-gratis/comprobado\\_190411?term=check&page=1&position=1&page=1&position=1&related\\_id=190411&origin=search](https://www.flaticon.es/icono-gratis/comprobado_190411?term=check&page=1&position=1&page=1&position=1&related_id=190411&origin=search)

[https://www.flaticon.com/free-icon/trash\\_2496733?term=trash&page=1&position=29&page=1&position=29&related\\_id=2496733&origin=search](https://www.flaticon.com/free-icon/trash_2496733?term=trash&page=1&position=29&page=1&position=29&related_id=2496733&origin=search)

- Fuente empleada en la página web:

Archivo .ttf descargado de: <https://www.dafont.com/es/louis-george-caf.font>

Página para convertir .ttf a woff: <https://convertio.co/es/ttf-woff/>

- Términos y servicios y política de privacidad de la página generados de:

<https://www.generateprivacypolicy.com/>

<https://www.termsofservicegenerator.net/>