
Calculadora Web con Reconocimiento de Voz

"Hacer cálculos nunca fue tan fácil"



TRABAJO FINAL DE CICLO

CFGS DESARROLLO DE APLICACIONES WEB

Autor/a: Laura Arellano Torrero

Tutor/a: Tomás Martínez Ruiz

Junio de 2025

ÍNDICE

Capítulo 1: Introducción y Objetivos	2
Capítulo 2: Especificación de Requisitos	4
Capítulo 2.1: Requisitos Funcionales	4
Capítulo 2.2: Requisitos No Funcionales	4
Capítulo 2.3: Bocetos/prototipos diseñados en Figma	5
Capítulo 3: Planificación Temporal y Evaluación de Costes	12
Capítulo 3.1: Planificación Temporal	12
Capítulo 4: Tecnologías utilizadas	14

Capítulo 1: Introducción y Objetivos

En este proyecto voy a desarrollar una calculadora web que se va a poder usar de dos maneras: como una calculadora normal, usando **botones** en la pantalla, y también usando la **voz** para decir las operaciones que queremos hacer.

La interacción mediante botones será similar a las calculadoras tradicionales, lo que ofrece una **experiencia de usuario familiar** a aquellos usuarios que prefieren usar métodos tradicionales.

Por otro lado, la opción de interactuar mediante voz se presenta como una solución innovadora que mejorará la experiencia del usuario, especialmente para quienes buscan una forma más **cómoda y rápida** de realizar operaciones.

La idea ha nacido de observar cómo cada vez más personas interactúan con sus dispositivos utilizando comandos de voz, lo cual no sólo aporta comodidad y rapidez, sino también **accesibilidad** para aquellos usuarios con dificultades visuales.

La aplicación tendrá dos modos de uso: uno **clásico**, mediante botones en pantalla, y otro **innovador**, en el que los usuarios podrán pronunciar las operaciones que desean resolver.

El objetivo principal es crear una aplicación web que sea lo más **intuitiva y accesible posible**, aprovechando las tecnologías actuales del desarrollo web. La **facilidad** de uso es una prioridad, y se espera que tanto el diseño como la interacción de la calculadora sean lo suficientemente **simples** para que cualquier persona pueda usarla sin dificultades. Para llevar a cabo este proyecto, se utilizará **Angular**, un framework basado en TypeScript.

Además, se implementará una **API de reconocimiento de voz**, que será la encargada de escuchar al usuario e interpretar sus palabras, traducéndose a operaciones matemáticas siempre y cuando el **habla** sea **clara, pausada y comprensible**, lo que permitirá una interpretación adecuada de las instrucciones dadas.

La calculadora se centrará exclusivamente en el funcionamiento en el lado del cliente (frontend), lo cual permite un enfoque más directo y centrado en el comportamiento **visual e interactivo** de la aplicación.

Calculadora Web con Reconocimiento de Voz
Laura Arellano Torrero
CFGS DESARROLLO DE APLICACIONES WEB.

Aunque el cliente de este proyecto será ficticio, he imaginado que se podría tratar, por ejemplo, de una **academia online** que necesita una herramienta sencilla y accesible para realizar cálculos rápidos en sesiones de clase, o incluso de una **persona mayor o con discapacidad motora** que necesita una **solución** que no dependa del uso constante del teclado o del ratón.

Este proyecto representa la oportunidad de **profundizar en tecnologías actuales** del desarrollo web, **experimentar** con herramientas modernas y trabajar con APIs externas, todo ello enfocado en crear una solución accesible y orientada al usuario.



Capítulo 2: Especificación de Requisitos

Antes de iniciar el desarrollo de la calculadora web, veo necesario plantear los **requisitos** que debe **cumplir** mi proyecto. A continuación, describiré los **requisitos funcionales** y **no funcionales**, y también adjuntaré **bocetos** del diseño de la calculadora realizados en **Figma**.

Capítulo 2.1: Requisitos Funcionales

Mi aplicación debe cumplir los siguientes requisitos funcionales para alcanzar los objetivos propuestos en el Capítulo 1:

- **Operaciones básicas:** La calculadora debe ser capaz de hacer sumas, restas, multiplicaciones y divisiones.
- **Entrada por voz:** El usuario debe poder dictar operaciones matemáticas sencillas a través de comandos por voz.
- **Visualización de resultados:** Después de cada operación, el resultado debe mostrarse claramente en la pantalla de la calculadora.
- **Activar/Desactivar el micrófono:** El usuario debe poder activar o desactivar el modo de reconocimiento de voz pulsando un botón.
- **Gestión de errores:** Si el **reconocimiento de voz** no entiende la orden del usuario, debe aparecer un **mensaje de error**.
- **Borrar datos:** Por último, la calculadora necesita tener la opción de **borrar el contenido** actual para poder hacer una **nueva operación**.

Capítulo 2.2: Requisitos No Funcionales

Los requisitos no funcionales describen **cómo** debe **comportarse** la aplicación:

- **Tiempo de respuesta:** La aplicación debe ser capaz de realizar los cálculos de manera **rápida y eficiente**, mostrando el resultado de las operaciones en pocos segundos. Esto es fundamental para **evitar** elevados **tiempos de espera** y mantener una **buena experiencia de usuario**.

CFGS DESARROLLO DE APLICACIONES WEB.

- **Compatibilidad:** La calculadora debe ser compatible con **el mayor número posible de navegadores**. Asegurar esta compatibilidad permitirá que la aplicación sea accesible para diferentes usuarios, sin importar el tipo de navegador que estén utilizando.
- **Diseño responsivo:** La interfaz debe adaptarse correctamente a diferentes **tamaños de pantalla**, ya sea en dispositivos móviles, tablets o en ordenadores. Un diseño responsivo mejorará la accesibilidad y la experiencia de uso.
- **Accesibilidad:** El diseño de la aplicación debe emplear colores de **alto contraste** y tamaños de letra adecuados. Esto ayudará a que la herramienta sea inclusiva y esté al alcance de cualquier persona.
- **Experiencia de Usuario (UX):** La aplicación debe ofrecer una interfaz intuitiva y sencilla pensada para usuarios de **todas las edades**. El objetivo es que cualquier persona, sin necesidad de conocimientos técnicos, pueda utilizar la calculadora de forma **rápida y sin complicaciones**.
- **Modo claro y modo oscuro:** La aplicación debe contar con la opción de cambiar entre modo claro y modo oscuro, permitiendo que cada usuario elija el **estilo visual** que más le guste.

Capítulo 2.3: Bocetos/prototipos diseñados en Figma

Antes de empezar a programar la calculadora, he diseñado los **bocetos** de la aplicación en Figma. Esta herramienta me ha permitido planificar **cómo** quiero que **se vea** mi aplicación web, qué **colores** se van a usar y cómo se **organizará cada parte** antes de pasar al desarrollo. Estos bocetos se han diseñado para adaptarse correctamente a diferentes tamaños de pantalla (**ordenador, tablet y móvil**), y para ofrecer dos modos visuales: **modo claro y modo oscuro**, que el usuario podrá alternar según sus preferencias.

A lo largo de este apartado, voy a mostrar las distintas páginas diseñadas: **Inicio, Cómo funciona y Accesibilidad**.

Características comunes de las **tres páginas** diseñadas:

Las tres páginas comparten la misma cabecera, que incluye el **logo** de la aplicación, un **menú de navegación** sencillo que permite navegar entre ellas y un **botón** de cambio de tema (**modo claro/oscuro**) representado por un icono dependiendo del modo que se esté utilizando en ese momento (un sol o una luna).

Calculadora Web con Reconocimiento de Voz

Laura Arellano Torrero

CFGS DESARROLLO DE APLICACIONES WEB.

En cada imagen, el **modo oscuro** se muestra en el lado **izquierdo** y el **modo claro** en el lado **derecho**.

En el diseño responsivo (para tamaño **small**), el **logo no es visible** y el **menú** de navegación se reduce a un **icono de tres líneas horizontales**.

Por último, en el **pie de página** aparece mi nombre, con el fin de proteger los **derechos de autor** correspondientes.

Página de Inicio:

Esta es una de las **páginas más importantes** de todo el proyecto, ya que no solo es la **primera impresión** que recibe el usuario al acceder a la web, sino que además **contiene directamente la calculadora**, que es el elemento principal y funcional del sitio.

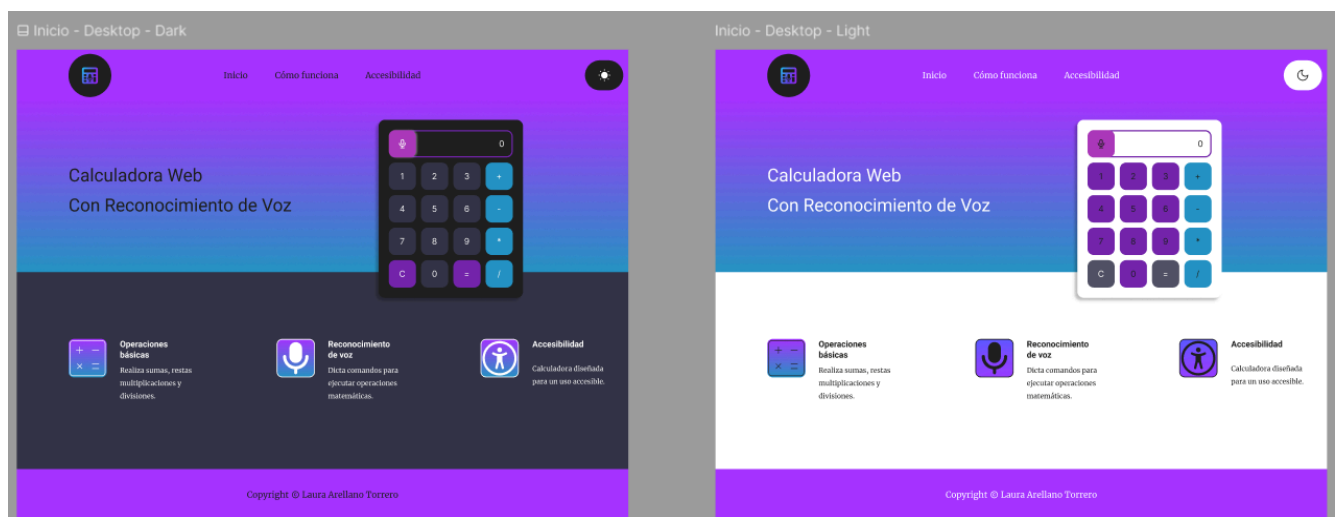
La calculadora se muestra en el **centro** de la pantalla, con un **diseño moderno** que destaca sobre el fondo. Desde aquí, el usuario puede empezar a hacer **operaciones**, ya sea pulsando los botones de los **números** y de los **operadores** o activando el **micrófono** para usar el modo de reconocimiento de voz.

Debajo de la calculadora he añadido **tres “secciones”** con iconos explicativos, que resumen las funcionalidades de mi calculadora:

- **Operaciones básicas** (sumas, restas, multiplicaciones y divisiones)
- **Reconocimiento de voz** para dictar operaciones
- **Accesibilidad** (como parte importante del diseño).

A continuación adjunto imágenes de la página de Inicio, diseñada en Figma y adaptada a **distintos tamaños**, en **ambos modos visuales**: oscuro y claro.

En la siguiente imagen aparece la página **Inicio** en tamaño **Desktop** con un ancho de **1440px**:

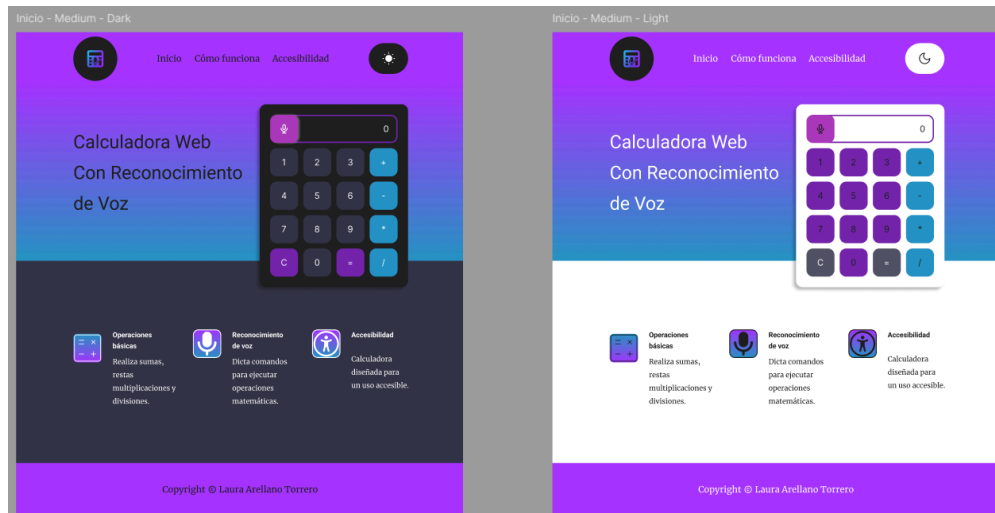


Calculadora Web con Reconocimiento de Voz

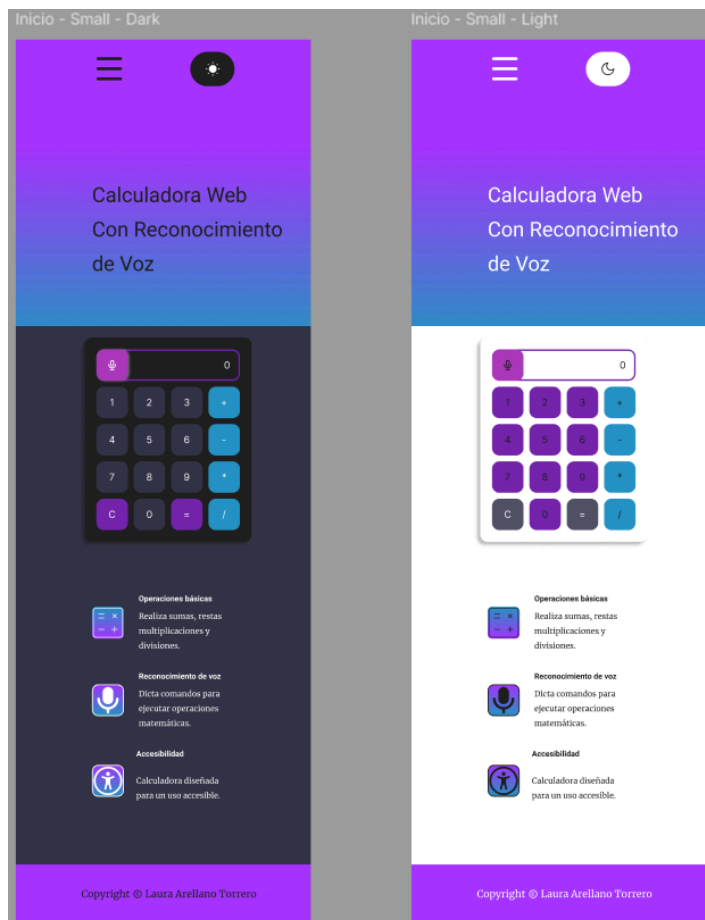
Laura Arellano Torrero

CFGS DESARROLLO DE APLICACIONES WEB.

A continuación, aparece la página de **Inicio** en tamaño **Medium** con un ancho de **1024px**:



Por último, en la siguiente imagen se puede ver la página de **Inicio** en tamaño **Small** con un ancho de **600px**:



Calculadora Web con Reconocimiento de Voz

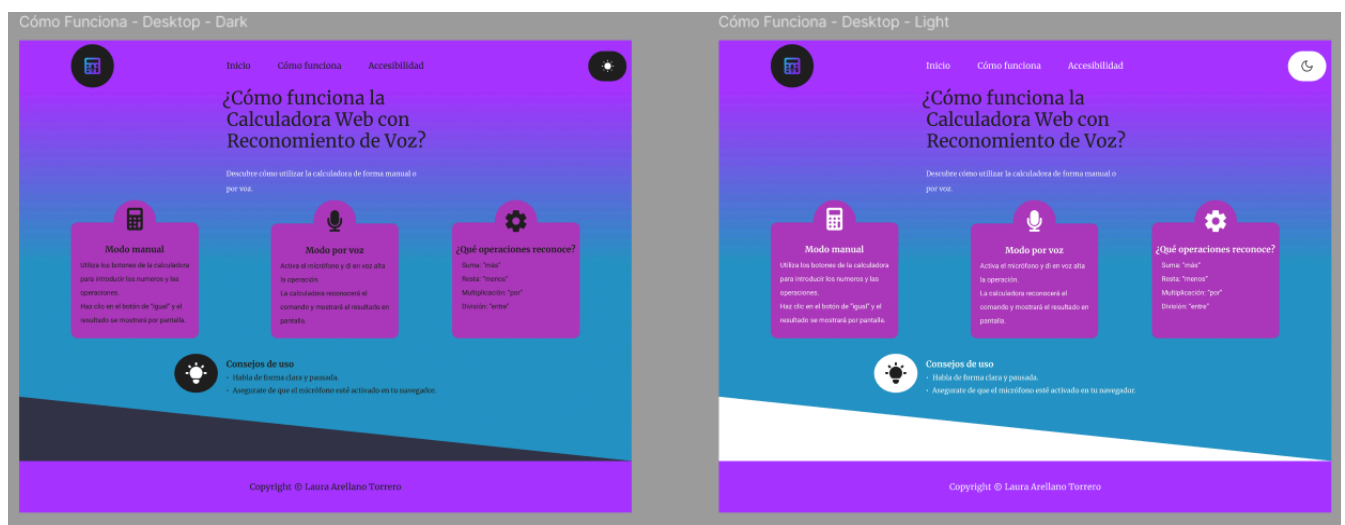
Laura Arellano Torrero

CFGS DESARROLLO DE APLICACIONES WEB.

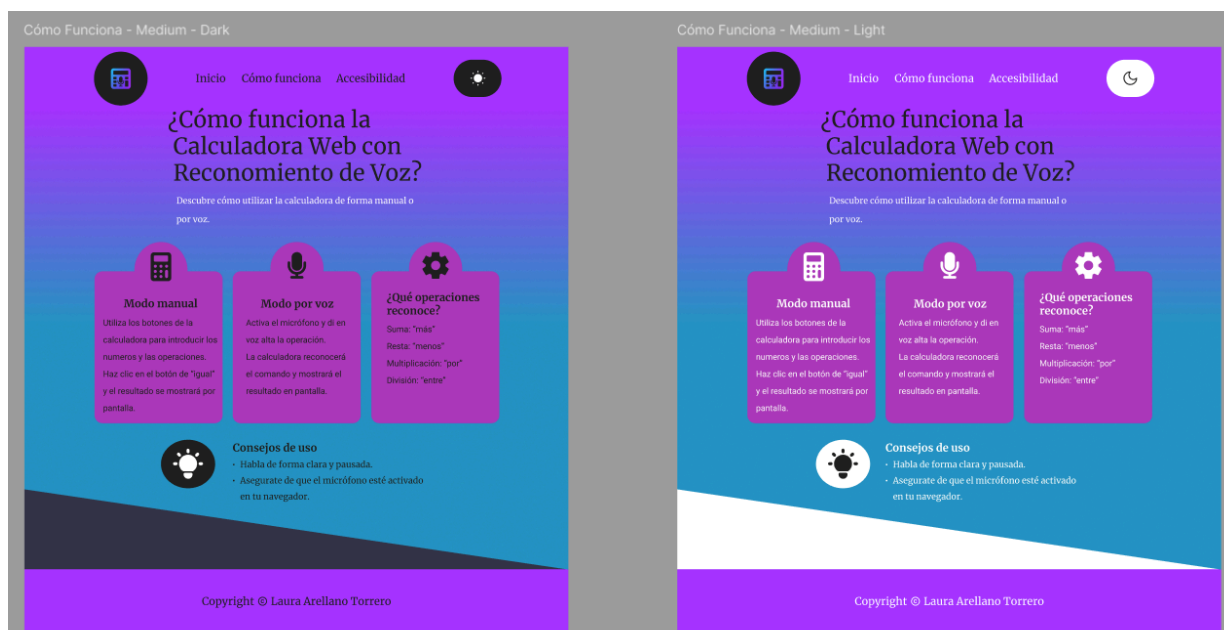
Página Cómo funciona:

Para ayudar al usuario en el uso de la aplicación, he diseñado **tarjetas informativas** en las que explico de forma clara el uso de la calculadora: desde el modo **manual** hasta el uso por **voz**, incluyendo una descripción de las **operaciones** que la calculadora **puede reconocer**.

En la siguiente imagen aparece la página “**Cómo funciona**” en tamaño **Desktop** con un ancho de **1440px**:



Ahora, aparece la página “**Cómo funciona**” en tamaño **Medium** con un ancho de **1024px**:



Calculadora Web con Reconocimiento de Voz

Laura Arellano Torrero

CFGS DESARROLLO DE APLICACIONES WEB.

Por último, en la siguiente imagen se puede ver la página “**Cómo funciona**” en tamaño **Small** con un ancho de **600px**:



Como se puede observar, en la página “Cómo funciona” he utilizado **iconos** y **textos cortos** descriptivos, favoreciendo una **lectura rápida e intuitiva**.

Este diseño ha sido pensado para explicar el uso de la aplicación lo más **fácil** posible, dando más importancia al **modo por voz**, que puede ser **menos común** para personas con **poca experiencia tecnológica**.

Página Accesibilidad:

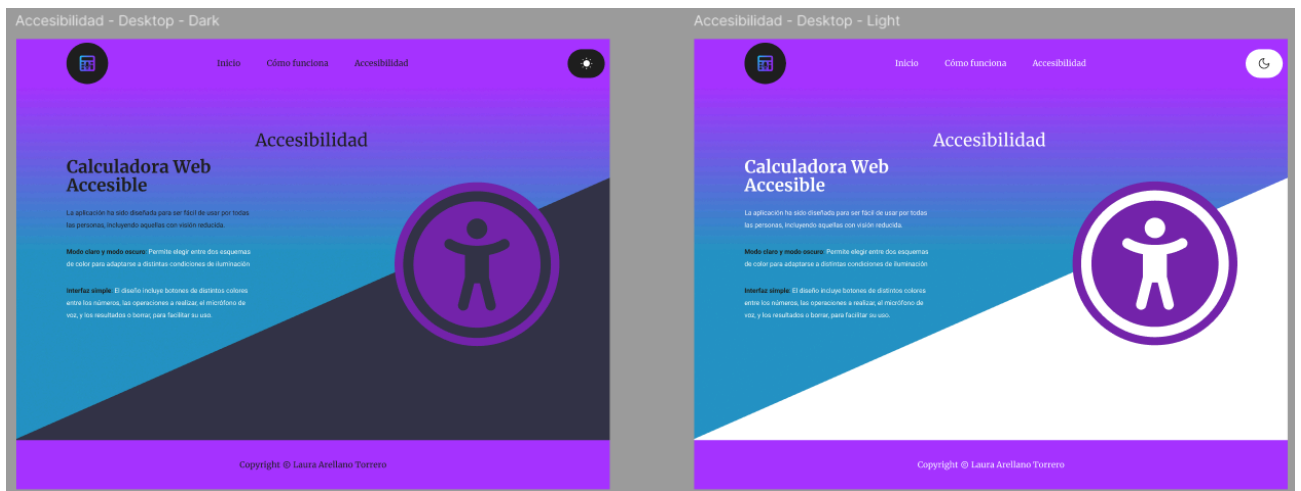
He diseñado esta página centrándome en la inclusión, con el objetivo de **garantizar** que cualquier persona, **independientemente** de sus **capacidades visuales** o su **experiencia digital**, pueda utilizar la aplicación fácilmente.

Calculadora Web con Reconocimiento de Voz

Laura Arellano Torrero

CFGS DESARROLLO DE APLICACIONES WEB.

En la siguiente imagen aparece la página “Accesibilidad” (Desktop) con un ancho de **1440px**:



A continuación, aparece la página “Accesibilidad” en tamaño **Medium** con un ancho de **1024px**:



Calculadora Web con Reconocimiento de Voz
Laura Arellano Torrero
CFGS DESARROLLO DE APLICACIONES WEB.

Por último, en la siguiente imagen se puede ver la página “**Accesibilidad**” en tamaño **Small** con un ancho de **600px**:



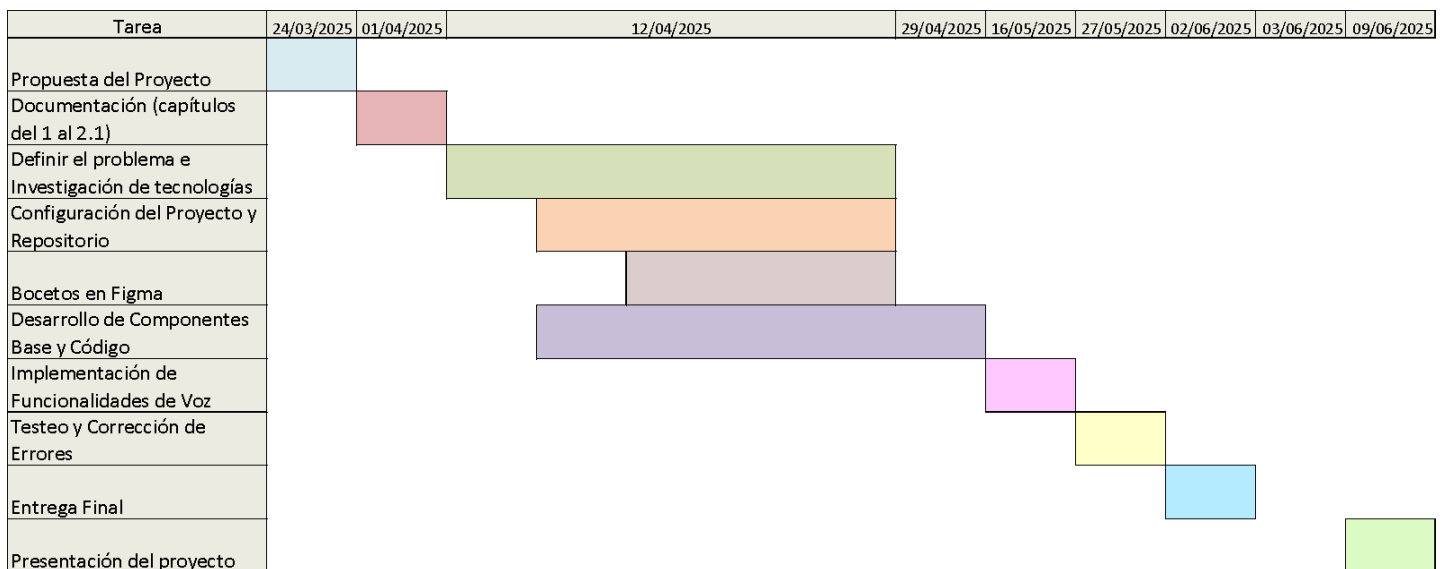
Como se ha podido observar en las imágenes presentadas para cada una de las páginas, el diseño se adapta bien tanto en modo claro como en modo oscuro, manteniendo un buen **diseño responsivo**. Esto asegura una buena **experiencia visual y accesible** sin importar el dispositivo desde que el usuario esté accediendo.

Capítulo 3: Planificación Temporal y Evaluación de Costes

Capítulo 3.1: Planificación Temporal

La planificación temporal de este proyecto tiene como objetivo **estructurar su desarrollo** en **fases ordenadas**, permitiendo **controlar** el progreso y **facilitando la entrega a tiempo**.

He elaborado un **diagrama de Gantt** en Excel que refleja las **tareas principales** a desarrollar, los **plazos previstos** y su relación con las **fechas de entrega**:



Esta planificación está dividida en **varias fases**, como **investigación, diseño, desarrollo, pruebas y entrega**, para organizar mejor el trabajo.

Aunque he establecido unos **plazos aproximados**, es posible que durante el desarrollo **surjan imprevistos** que obliguen a hacer pequeños cambios.

El diagrama de Gantt me sirve para visualizar de forma rápida **qué tareas** tengo que realizar en cada momento y **cuánto tiempo** tengo disponible. También he dejado algo de **margen** entre fases **por si alguna tarea lleva más tiempo** del esperado, intentando así **evitar retrasos en la entrega final**.

Capítulo 3.2: Evaluación de Costes

Antes de empezar con el desarrollo de cualquier proyecto web, es importante hacer una estimación de los costes para saber qué recursos serían necesarios si este proyecto se hiciera de forma profesional.

PLANIFICACIÓN DE HORAS DE TRABAJO:

Para poder **organizar** mejor el proyecto y la **evaluación de los costes**, he hecho una **estimación** de las horas que me llevaría completar cada parte del desarrollo.

Tarea	Horas estimadas
Investigación de tecnologías y planificación	10 horas
Diseño de prototipos en Figma	35 horas
Configuración del proyecto Angular y estructura básica	20 horas
Desarrollo de componentes y funcionalidades principales	40 horas
Integración de reconocimiento de voz	5 horas
Estilos SCSS y diseño responsivo	15 horas
Pruebas, corrección de errores...	5 horas
Documentación	15 horas

Calculadora Web con Reconocimiento de Voz
Laura Arellano Torrero
CFGS DESARROLLO DE APLICACIONES WEB.

A continuación, se puede ver una evaluación aproximada de los costes que tendría el desarrollo de mi aplicación web:

EVALUACIÓN DE COSTES

Concepto	Descripción	Coste estimado (€)
Dominio web	Registro de un dominio .com, .es o parecido	12€/año
Alojamiento web (Hosting)	Plan básico de alojamiento.	50€/año
Diseño (Figma)	Versión gratuita. Si desea la versión PRO, tendrá coste.	0€
Framework Angular	Angular es de código abierto y gratuito	0€
API Reconocimiento de Voz	Web Speech API; no requiere coste.	0€
Control de versiones (GitHub)	GitHub ofrece plan gratuito para proyectos individuales	0€
Visual Studio Code	Software libre. Sin coste.	0€
Horas de desarrollo	Un total de 145 horas de trabajo, 12€/hora.	145 horas x 12€ = 1.740€
Costes adicionales	Imprevistos (actualizaciones, pruebas, corrección de errores...)	20€
TOTAL		1.822€

Capítulo 4: Tecnologías utilizadas

En este proyecto he elegido varias **tecnologías** que me han parecido las más **adecuadas** para llevar a cabo el proyecto. A continuación, explico las **principales tecnologías** que estoy utilizando, por qué las he elegido y cómo van a ayudar a que la aplicación funcione de manera correcta.

Lenguajes de programación:

- **HTML**: **HTML** es ideal para **estructurar contenido** en la web, y es compatible con la mayoría de navegadores. Al ser tan común y fácil de usar, me permite crear la **estructura básica de la página** de manera clara y ordenada. Esto incluye la parte de la **calculadora**, los **botones**, el **área para mostrar resultados** y los **campos de entrada** de las operaciones.
- **SCSS (Sass)**: En lugar de usar CSS tradicional, he trabajado con **SCSS**, que es una versión mejorada de CSS. Este lenguaje me permite **escribir estilos** de forma más **limpia y organizada**, usando **variables**, **funciones** o **anidación de reglas**. Esto hace que el **código** sea mucho **más fácil de mantener**, sobre todo cuando el proyecto empieza a crecer. **Angular** además **lo soporta muy bien**, así que es perfecto para este tipo de aplicaciones web.
- **Javascript**: **Javascript** es el lenguaje que hace posible que la calculadora sea **interactiva**. Me permite implementar todas las funcionalidades que necesita la aplicación, como hacer los **cálculos** y **recibir comandos de voz**.

Framework:

- **Angular (TypeScript)**: **Angular** es un framework **súper completo y fácil de usar**, ideal para crear aplicaciones web interactivas.

Lo que más me gusta de Angular es que me permite **organizar todo el código** de manera clara y estructurada, **usando componentes**. Además, al usar **TypeScript**, puedo trabajar con un **sistema de tipos más seguro** que me ayuda a **evitar errores** a la hora de escribir el código. Esto es muy **útil** para **proyectos grandes y complejos**, y me da más control sobre el desarrollo.

API de Reconocimiento de Voz:

- Web Speech API: La **Web Speech API** es la herramienta perfecta para integrar el **reconocimiento de voz**. Gracias a esta API, los usuarios pueden **decir** las **operaciones matemáticas** en lugar de escribirlas, lo que hace que la calculadora sea mucho **más accesible**. Esta API convierte la voz en texto, lo cual me permite interpretar los **comandos de voz y convertirlos en operaciones matemáticas**.

Control de versiones y repositorio:

- Git y GitHub: **Git** es una herramienta que permite llevar un **control total del código** y de los **cambios** que se van realizando a lo largo del desarrollo.

Al usar **GitHub** como **repositorio** en la nube, puedo **almacenar el proyecto** de forma segura y acceder a él desde cualquier lugar. Además, GitHub me permite mantener el proyecto **bien organizado**, ya que puedo hacer **cambios** sin miedo a perder el trabajo anterior. Si algo no funciona, está la opción de **volver a una versión anterior** del código. También permite hacer un seguimiento de la **evolución del proyecto** y colaborar con otras personas si fuera necesario.

Diseño de la interfaz de usuario (UI):

- Figma: **Figma** es una **herramienta de diseño** muy popular y fácil de usar para crear **prototipos o bocetos** de interfaces web.

Decidí usar Figma para diseñar los bocetos de mi calculadora porque es muy **práctica** y me permite ver cómo quedará la interfaz antes de empezar a programarla. Además, Figma me permite **crear variables** que luego se pueden **exportar** para usar fácilmente a la hora de programar.

Base de Datos:

Este proyecto **no necesita base de datos** porque el funcionamiento es del lado del cliente. La calculadora **no guarda datos**, simplemente resuelve operaciones. Si en un futuro se quisiera añadir la opción de **guardar historial de operaciones** o **preferencias** del usuario, se podría usar una base de datos como **Firestore, MongoDB...**

Otras herramientas y tecnologías:

- NPM (Node Package Manager): NPM es la herramienta que ayuda a gestionar todas las librerías y dependencias del proyecto. También, facilita la instalación y actualización de todas estas librerías de manera rápida y sencilla.