

EyeSee: La aplicación que identifica con quien te cruzas

Memoria del proyecto

Trabajo de Fin de Grado
Grado en Ingeniería Informática



VNiVERSiDAD
D SALAMANCA

Junio 2023

Autor

Silvia Requejo López

Tutor/a

Gabriell Villarrubia González

André Sales Mendes

Certificado de tutores

D. André Sales Mendes y D. Gabriel Villarrubia González, profesores del Departamento de Informática y Automática de la Universidad de Salamanca,

HACEN CONSTAR:

Que el trabajo titulado “EyeSee: La aplicación que identifica con quién te cruzas”, que se presenta, ha sido realizado por Dña. Silvia Requejo López, con DNI ****4947G y constituye la memoria del trabajo realizado para la superación de la asignatura Trabajo de Fin de Grado en la Titulación Grado en Ingeniería Informática en esta Universidad.

Salamanca, 4 de julio de 2023.

Resumen

El proyecto realizado como Trabajo de Fin de Grado consiste en el diseño y la implementación de una aplicación web que tendrá como usuarios finales personas con un grado de discapacidad visual.

La aplicación les permitirá saber si una persona conocida se está aproximando a ellos mientras estén dando un paseo. Para ello, los usuarios podrán guardar a sus amigos en su perfil, donde la aplicación les solicitará que suban una foto para identificar a cada uno de ellos. De este modo, una vez haya configurado su perfil el usuario podrá iniciar paseos.

Los paseos se iniciarán desde la aplicación web, aunque también será necesario que el usuario ejecute la tecnología de reconocimiento facial desde un miniordenador que portará durante el camino. Para cada paseo, se guardará la duración y si un amigo guardado se aproximara al cliente, la aplicación le avisaría, para que supiera de antemano con quién se va a encontrar.

Este sistema pretende facilitar el día a día de personas con discapacidad visual, reduciendo brechas de desigualdad existentes.

Durante todo el proyecto se han seguido las prácticas aprendidas en las asignaturas de Ingeniería del Software del grado. Así mismo, se ha elaborado la documentación necesaria para todas las tareas que se han llevado a cabo.

Palabras clave: Discapacidad visual, Paseos, Calle, Reconocimiento facial, Deep Learning.

Abstract

The project carried out as an End of Degree Project consists of the design and implementation of a web application that will have as end users people with a degree of visual disability. In this way, the application will allow users to know if a friend or an acquaintance is approaching them while they are outside taking a walk.

Thus, users will be able to save friends in their profile just by uploading a photo to identify each of them. In this way, once the profile has been set up the user can start employing it for their everyday walks.

The web application will instantly start recording the walk, although it will also be necessary for the user to start the facial recognition technology from a mini-computer that he must carry in his walk. The duration of each exercise will be recorded into the app, and if a saved friend approaches the user, the application will warn him in loud voice, so that they know in advance who to greet.

This system aims to facilitate the daily life of those people with visual disabilities, thus reducing existing inequality gaps. Throughout the project, the practical knowledge and skills acquired in the subjects of Software Engineering has been applied. The necessary documentation has also been prepared for all the tasks carried out.

Keywords: Visual disability, Walks, Street, Facial recognition, Deep learning.

ÍNDICE

Certificado de tutores	2
Resumen	4
Abstract	6
ÍNDICE	8
ÍNDICE DE FIGURAS	10
ÍNDICE DE TABLAS	12
1. Introducción.....	12
2. Objetivos	17
2.1 Objetivos del sistema	17
2.2 Objetivos personales.....	18
3. Conceptos teóricos	19
3.1 Reconocimiento facial	19
3.2 Redes neuronales	22
3.3 Text to speech (TTS).....	29
4. Técnicas y herramientas	31
4.1 Herramientas de desarrollo web	32
4.1.1 JavaScript.....	32
4.1.2 HTML.....	32
4.1.3 CSS	33
4.1.4 Vue.js.....	33
4.1.5 Vuetify.....	34
4.1.6 NPM.....	34
4.1.7 Firebase.....	34
4.2 Herramientas para el reconocimiento facial	35
4.2.1 Python	35
4.2.2 Pytorch	36
4.3 Entorno de desarrollo	37
4.4 Herramientas CASE	37
4.4.1 Visual paradigm.....	38
4.4.2 EZEstimate	38
4.4.3 REM.....	39
4.4.4 Microsoft Project.....	39

4.4.5	Microsoft Word.....	40
4.4.6	Adobe XD	40
5.	Aspectos relevantes.....	41
5.1	Esquema general del sistema	41
5.2	Marco de trabajo	41
5.3	Estimación temporal.....	43
5.4	Planificación temporal	44
5.5	Especificación de requisitos	46
5.5.1	Objetivos del sistema.....	49
5.5.2	Requisitos de información	50
5.5.3	Requisitos no funcionales.....	51
5.5.4	Actores	51
5.5.5	Casos de uso.....	51
5.5.6	Diagramas de casos de uso	53
5.6	Análisis de requisitos.....	53
5.6.1	Modelo del dominio.....	54
5.6.2	Casos de uso del análisis	55
5.7	Diseño del sistema	55
5.7.1	Patrones arquitectónicos.	56
5.7.2	Subsistemas de diseño.....	57
5.7.3	Clases de diseño	58
5.7.4	Realización de casos de uso de diseño.....	59
5.7.5	Diseño de la base de datos	59
5.7.6	Implementación	60
5.7.7	Pruebas	61
5.8	Reconocimiento facial	62
5.8.1	Investigación previa	62
5.9	Diseño de la aplicación	64
6.	Conclusiones.....	73
7.	Líneas futuras	74
8.	Referencias.....	75

ÍNDICE DE FIGURAS

Figura 1. Discapacidad visual en personas con una edad superior a 6 años. Extraído de [1]	13
Figura 2. Cifras de discapacidad visual según el INE [1]	14
Figura 3. Técnicas de detección de rostro. Extraído de [4]	21
Figura 4. Reconocimiento facial. Extraído de [5]	22
Figura 5. Esquema de una red neuronal. Extraído de [6]	23
Figura 6. Explicación del proceso de entrenamiento. Extraído de [7]	24
Figura 7. RNN. Extraído de [9]	26
Figura 8. Operación de convolución. Extraído de [10]	27
Figura 9. Función ReLU. Extraída de [10]	27
Figura 10. Capa de agrupación. Extraído de [10]	28
Figura 11. CNN. Extraído de [12]	28
Figura 12. Estructura de un TTS. Extraído de [13]	29
Figura 13. Codificador. Extraído de [13]	30
Figura 14. Decodificador. Extraído de [13]	31
Figura 15. Pytorch. Extraído de [19]	36
Figura 16. Visual Studio Code	37
Figura 17. Visual Paradigm	38
Figura 18. EZEstimate	39
Figura 19. Ciclo de vida del proceso unificado. Extraído de [20]	43
Figura 20. Estimación realizada con EZEstimate	44
Figura 25. Planificación iteración 5	45
Figura 31. Diagrama de Gantt para la fase de construcción	46
Figura 36. Diagrama de casos de uso del paquete paseos	47
Figura 37. Diagrama de actores	48
Figura 38. Modelo del dominio	54
Figura 39. Añadir detalles a un paseo	55
Figura 41. MVVM	56
Figura 42. Patrón Singleton	57
Figura 43. Subsistemas de diseño	57
Figura 44. Paquete Vista	58
Figura 45. Paquete Modelo de vista	58

Figura 46. Paquete modelo	59
Figura 47. Diagrama de secuencia.....	59
Figura 48. Diagrama de clases de Firestore.....	60
Figura 49. Registro	65
Figura 50. Inicio de sesión.....	66
Figura 51. Recuperación de la contraseña.....	66
Figura 52. Pantalla de inicio	67
Figura 53. Pantalla mis amigos	68
Figura 54. Añadir amigo	68
Figura 55. Eliminar amigo.....	69
Figura 56. Página principal paseos.	69
Figura 57. Añadir descripción.....	70
Figura 58. Encuentro producido	71
Figura 59. Pantalla de configuración.	71
Figura 60. Modificación de contraseña.....	72

ÍNDICE DE TABLAS

Tabla 1. Participante Silvia Requejo López.....	48
Tabla 2. Gestión de paseos.....	49
Tabla 3. Información sobre encuentros.....	50
Tabla 4. Actor usuario	51
Tabla 5. Registrar encuentro.....	53

1. Introducción

La tecnología resulta imprescindible hoy en día en nuestra sociedad. En general, los avances producidos en los últimos años tratan de conseguir mayor bienestar y comodidad para los usuarios de dichas tecnologías y gran parte de

la sociedad se beneficia sin ninguna dificultad de todos estos avances. Sin embargo, existen sectores de la sociedad que, por un motivo u otro, necesitan que la tecnología se adapte a ellos para poder usarla.

En este proyecto, nos hemos centrado en las personas con discapacidad visual. En España, la última encuesta realizada por el Instituto Nacional de Estadística (INE) a cerca de personas con discapacidad visual data de 2020. Si consultamos los datos de esa encuesta en la página web del INE (<https://www.ine.es/dynt3/inebase/es/index.htm?padre=8494&capsel=8754>), podemos observar las últimas cifras recogidas en nuestro país. En este caso, buscaremos por tipo de discapacidad las cifras nacionales por sexo y edad, seleccionando todos los sexos y todas las edades. La consulta realizada se muestra en la figura 1.

Discapacidad. Cifras absolutas

Discapacidades y deficiencias

Población de 6 y más años con discapacidad según grupo de discapacidad por sexo y edad.

Unidades: miles de personas

▶ Seleccione valores a consultar

Sexo	Edad	Grupo de discapacidad
Total	Total	Total
Hombre	De 6 a 15 años	Visión
Mujer	De 16 a 24 años	Audición
	De 25 a 34 años	Comunicación
	De 35 a 44 años	Aprendizaje, aplicación del conocimiento y desarrollo de tareas
	De 45 a 54 años	Movilidad

Seleccionados: 3 Total: 3 Seleccionados: 1 Total: 13 Seleccionados: 1 Total: 9

▶ Elija forma de presentación de la tabla

Sexo	Edad	Grupo de discapacidad

Decimales a mostrar: Por defecto

▶ Notas

1) Una persona puede tener discapacidades de más de un grupo de discapacidad

Figura 1. Discapacidad visual en personas con una edad superior a 6 años. Extraído de [1]

En la figura 2 se muestran los resultados de la consulta, como podemos observar, se estiman un total de más de un millón de personas con alguna discapacidad visual en España.

Discapacidad. Cifras absolutas

Discapacidades y deficiencias

Población de 6 y más años con discapacidad según grupo de discapacidad por sexo y edad.

Unidades: miles de personas

Tabla		Gráfico
		Visión
Total		
Total		1.051,3
Hombre		
Total		400,4
Mujer		
Total		650,9

Figura 2. Cifras de discapacidad visual según el INE [1]

Ha de tenerse en cuenta que en todas estas estadísticas se incluyen todos los grados de discapacidad visual, sin embargo, se deben de hacer matices. El Real decreto 888/2022 establece los criterios de clasificación para las deficiencias permanentes del sistema visual. Según establece la tabla 4.2 del Capítulo 4 del citado decreto, podemos dividir las deficiencias visuales en: leves, moderadas, graves y profundas. Existe otra categoría que ya no se incluye como deficiencia visual y es la ceguera o casi ceguera, dentro de esta categoría podemos distinguir entre la casi ceguera y la ceguera total. Esta última subcategoría implica que la persona que padece esta discapacidad no percibe la luz. Estas categorías se definen según la escala de agudeza visual (EAV).

El objetivo principal de la aplicación web es conseguir que estas personas obtengan un beneficio en su día a día, consiguiendo una situación de igualdad de condiciones en sus encuentros.

Consideremos un paseo cualquiera en el que una persona que no padece ningún grado de discapacidad visual se encuentra a personas conocidas o amigos y los saluda con naturalidad. Supongamos ahora que la persona que se encuentra por la calle sí padece una discapacidad visual. En este caso, el conocido tendrá que acercarse a él, identificarse y quizá realizar algún tipo de contacto físico para ser reconocido. Si la persona con discapacidad visual fuera paseando con la aplicación con un paseo iniciado, sería avisado de que se le

Memoria. EyeSee

está aproximando un conocido, lo cual evitaría ese incómodo momento descrito anteriormente y ayudaría a reducir la desigualdad en un ámbito cotidiano, que se puede producir en múltiples ocasiones. El fin de la aplicación web desarrollada en este Trabajo de fin de grado es precisamente utilizar la tecnología para ayudar a los usuarios y lograr una mayor comodidad en su día a día.

El sistema desarrollado se conocerá con el nombre EyeSee y permitirá a un usuario crearse una cuenta, donde al registrarse tendrá que introducir sus datos personales.

Una vez la cuenta haya sido creada, la aplicación permitirá añadir amigos, que serán identificados mediante su nombre y apellidos; un alias, que se utilizará para referirse a ellos en caso de un encuentro y una descripción que permite especificar detalles sobre esa persona. Para cada amigo, el usuario ha de subir a la web una foto de este en el que se le vea la cara de forma reconocible. Dichas imágenes serán subidas a la base de datos y las utilizará la tecnología de reconocimiento facial durante los paseos.

El usuario puede iniciar un paseo en la aplicación web, después de asegurarse de activar la tecnología de reconocimiento facial. El paseo guardará la duración de este y si la tecnología de reconocimiento facial detectara un encuentro, el usuario sería avisado mediante un cuadro de texto que será leído en voz alta. Cuando la persona lo desee, podrá finalizar el paseo, que se guardará en su cuenta incluyendo los datos de duración, amigos encontrados y una breve descripción que podrá añadir manualmente cuando desee.

Por último, la aplicación web ofrecerá un apartado de configuración, donde se podrá modificar la información relativa al usuario.

La tecnología de reconocimiento facial se ejecutará en un miniordenador que la persona portará consigo durante los paseos. Dicho dispositivo será identificado por el usuario dentro de la app mediante su dirección MAC y será el encargado de ejecutar el script de reconocimiento facial.

La tecnología utilizada ha sido extraída de una librería de github: face-recognition [3] y adaptada a las necesidades de la aplicación web. Esta librería

Memoria. EyeSee

ofrece una API de reconocimiento facial, que cuenta con un sistema de reconocimiento facial ya entrenado.

En este documento se detallarán los aspectos más importantes del desarrollo del proyecto. La estructura del documento se basará en los siguientes puntos clave:

- Objetivos: se definirán las metas a lograr con el desarrollo de la aplicación web.
- Conceptos teóricos: se definirán los puntos teóricos clave en los que se ha basado el desarrollo y su relevancia para el trabajo realizado.
- Técnicas y herramientas: se describirán las técnicas y herramientas utilizadas en el desarrollo del proyecto.
- Aspectos relevantes del proyecto: se describen los aspectos más relevantes del sistema desarrollado.
- Conclusiones: se detallarán los resultados extraídos del proyecto.
- Líneas futuras: se explorarán posibles líneas de desarrollo para mejorar el proyecto actual.
- Referencias.

Este documento se encuentra completado con una serie de anexos que incluirán la documentación del proyecto y otros aspectos relevantes tratados en mayor profundidad.

- Anexo I. Planificación temporal y estimación del esfuerzo. Recoge las especificaciones temporales del proyecto y el esfuerzo estimado.
- Anexo II. Especificación de requisitos. Recoge los requisitos del sistema.
- Anexo III. Análisis de requisitos. Incluye la documentación de la fase de análisis.
- Anexo IV. Diseño del sistema. Incluye la documentación de la fase de diseño.
- Anexo V. Documentación técnica. Recoge la documentación de especificaciones técnicas del código de la aplicación web.

- Anexo VI. Manual de usuario. Recoge las funcionalidades del sistema explicadas para el usuario.

2. Objetivos

En este apartado se expondrán los objetivos perseguidos en el desarrollo del sistema EyeSee. Se detallarán dos tipos de objetivos, en primer lugar, los objetivos técnicos del sistema y en segundo lugar los objetivos personales.

2.1 Objetivos del sistema

El objetivo principal del sistema es la creación de una plataforma que permita a los usuarios detectar si una persona conocida se está acercando a ellos para saludarles durante un paseo.

Los objetivos técnicos que debe cumplir el sistema son:

- Gestión de usuarios. El sistema debe de gestionar de forma adecuada los diferentes usuarios de este, incluyendo los siguientes aspectos: alta, baja y modificación de datos personales de los usuarios.
- Gestión de amigos. El sistema ha de poseer los recursos suficientes para llevar a cabo la gestión completa de los amigos de los usuarios, incluyendo la carga de imágenes para los amigos, el alta de amigos, la modificación de datos personales y la eliminación de los mismos.
- Gestión de paseos. El sistema ha de ofrecer los servicios necesarios para gestionar los paseos de los usuarios incluyendo la duración de estos en tiempo real, la fecha y la descripción. También ha de ofrecer las opciones de inicio de un paseo y eliminación de un paseo ya existente. Dentro de la gestión de paseos encontramos un subobjetivo.
 - o Gestión de encuentros. El sistema ha de gestionar los encuentros producidos durante un paseo para mostrar en tiempo real un aviso en la aplicación web, también es necesario que los amigos que se han encontrado queden asociados a ese paseo, para que

posteriormente pueda ser recordado por el cliente de la aplicación.

- Gestión de la autenticación. El sistema ha de asegurarse de la fiabilidad del sistema de autenticación de usuarios, incorporando funcionalidades como el cierre de sesión, la eliminación de las cuentas, la seguridad de las contraseñas y la recuperación, en caso necesario, de estas últimas.

2.2 Objetivos personales

En este subapartado se detallan los objetivos personales de la autora. El objetivo principal para este proyecto ha sido la aplicación de la tecnología para la ayuda en la vida cotidiana de las personas que padecen discapacidad visual. Personalmente, me siento concienciada con este tema, pues en mi entorno cercano hay personas que tienen que lidiar a diario con las dificultades derivadas de una deficiencia visual. Por ello, he tratado de desarrollar esta aplicación para mejorar la calidad de vida en algo que, para personas que no cuentan con este problema es un detalle nimio, mientras que para los usuarios de esta app puede marcar la diferencia.

A su vez, el desarrollo de este proyecto ha supuesto la puesta en práctica de muchos de los conocimientos adquiridos durante el grado en varias asignaturas, como pueden ser: Ingeniería de software, Programación, Estructuras de datos, Bases de datos, Diseño de interfaces gráficas de usuario y Fundamentos de sistemas inteligentes, entre otras.

A mayores, este proyecto ha supuesto una gran oportunidad personal para ampliar mis conocimientos, ya que he aprendido nuevos lenguajes de programación como Python y JavaScript. También he utilizado el framework Vue.js, que me ha aportado mucho a la hora de desarrollar aplicaciones web de forma ágil y por último, he adquirido un conocimiento extenso acerca del servidor Firebase de Google. En resumen, esta ha sido mi primera oportunidad de desarrollar un proyecto de principio a fin de forma completa.

3. Conceptos teóricos

En este apartado se introducirán los principales conceptos teóricos necesarios para el desarrollo de la aplicación web. Los conceptos que se explicarán van a ser:

- Reconocimiento facial
- Redes neuronales
 - o Redes neuronales convolucionales
- Text to speech.

3.1 Reconocimiento facial

Un software de reconocimiento facial es aquel que confirma la identidad de una persona a partir de su rostro. Para ello es necesario que en primer lugar se produzca otro proceso, el de detección facial.

El proceso de detección trata de identificar si en una imagen dada existe o no un rostro y si existe cuál es su posición. Para un ordenador, esta puede ser una tarea muy compleja ya que se enfrenta a complicaciones como variaciones en la escala, la localización, la orientación, la expresión facial, las condiciones de brillo, etc.

Todas estas variables no afectan en la tarea de detección facial para un ser humano, ya que es una misión que encontramos muy sencilla. Sin embargo, para un ordenador un cambio en cualquiera de estas condiciones puede suponer todo un reto.

A continuación, mencionaremos los principales obstáculos en el proceso de detección facial. Estos motivos pueden hacer que se reduzca la exactitud de los resultados y la ratio de resultados obtenidos, pues recordemos que un ordenador desempeñando tareas de reconocimiento facial no sabrá previamente cuántos rostros habrá de encontrar en la imagen.

Principales retos:

Memoria. EyeSee

- Expresiones extrañas: los rostros humanos pueden adoptar expresiones distintas, las cuales pueden entorpecer el desarrollo de la tarea.
- Oclusión facial: durante el proceso de detección, la cara puede encontrarse parcialmente oculta, si por ejemplo la persona lleva gafas, mascarilla, se encuentra detrás de un objeto, etc. Esto también dificultará la detección facial.
- Brillo: valores de brillo muy alto o un gran contraste de iluminación entre diferentes partes de la imagen pueden suponer una gran dificultad en esta tarea.
- Fondo complejo: si la imagen a analizar tiene un fondo con muchos objetos, podría disminuir la tasa de detección.
- Demasiados rostros en la imagen: si la imagen contiene un gran número de rostros, la tarea de detección puede verse entorpecida.
- Baja resolución: las imágenes de poca calidad pueden ser más difíciles de analizar.
- Cambios en el tono de piel debidos a zonas geográficas: la piel de los humanos toma distintos tonos a veces según la zona geográfica en la que nos encontremos. Algunos de los métodos de detección facial utilizan los esquemas de color para detectar rostros, lo cual puede entorpecer la tarea, ya que estos esquemas de color en ningún caso van a ser fijos.
- Distancia: la distancia a la que se encuentra la cara a detectar respecto a la cámara influye, ya que a mayor distancia puede ser más difícil de detectar.
- Orientación: el ángulo con el cual es fotografiada la cara también influirá, ya que podría deformarse mucho el rostro y complicar la tarea.

En la figura 3 se pueden observar las diferentes técnicas utilizadas en la detección de rostros.

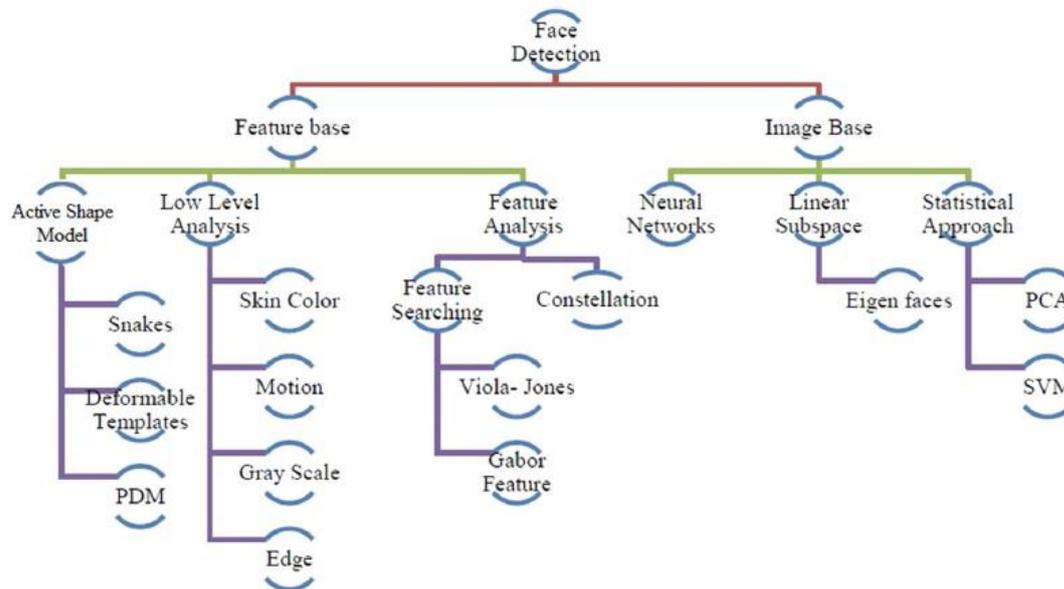


Figura 3. Técnicas de detección de rostro. Extraído de [4]

Principalmente, el problema de detección facial se puede enfocar desde dos ramas. La primera es la basada en características, que trata de extraer los rasgos de la imagen y los compara con el conocimiento de rasgos faciales. La segunda es el enfoque basado en imágenes, que trata de hallar coincidencias en la imagen, basándose en un entrenamiento previo.

Una vez se ha producido el proceso de detección facial, pasaremos al proceso de reconocimiento facial. El reconocimiento facial es el método que permite identificar la identidad de una persona a partir de su rostro. Este proceso es muy utilizado en imágenes o videos para detectar si la cara que aparece en dos imágenes distintas pertenece a la misma persona o para buscar un rostro dentro de una colección de imágenes. Así mismo, esta tecnología tiene su aplicación en los sistemas biométricos, que permiten identificar a usuarios o reforzar la seguridad de un sistema.

Un sistema de reconocimiento facial sigue los siguientes pasos. En primer lugar, realiza la detección ya explicada. En segundo lugar, analizará el rostro detectado y asignará la geometría y las expresiones faciales, identificando la

Memoria. EyeSee

ubicación de puntos de referencia clave como la distancia entre los ojos, la longitud entre la frente y la barbilla, entre la nariz y la boca, la profundidad de las cuencas oculares, la forma de los pómulos, etc.

Esos datos se convertirán a una cadena de coordenadas denominada huella facial, que identificará de forma unívoca a una persona.

El reconocimiento facial es capaz de identificar a una persona comparando su foto con otras imágenes y evaluar la probabilidad de que coincidan.

The screenshot displays the Amazon Rekognition 'Facial analysis' interface. It features a navigation menu on the left with categories like 'Custom Labels', 'Demos', 'Video Demos', 'Metrics', and 'Additional Resources'. The main content area shows a family photo with bounding boxes around the faces. Below the photo are two options: 'Choose a sample image' and 'Use your own image'. The 'Results' panel on the right, highlighted with a red border, provides a detailed analysis of the selected face.

Attribute	Confidence Score
looks like a face	99.9 %
appears to be male	99.9 %
age range	56 - 64 years old
smiling	96.2 %
appears to be happy	99.9 %
not wearing glasses	97.6 %

Figura 4. Reconocimiento facial. Extraído de [5]

3.2 Redes neuronales

Las redes neuronales son modelos simples que tratan de simular el funcionamiento del sistema nervioso. Están formadas por capas, que a su vez están compuestas por la unidad más básica, las neuronas. En la figura 5 se puede ver un esquema básico del funcionamiento de una red neuronal que explicaremos a continuación.

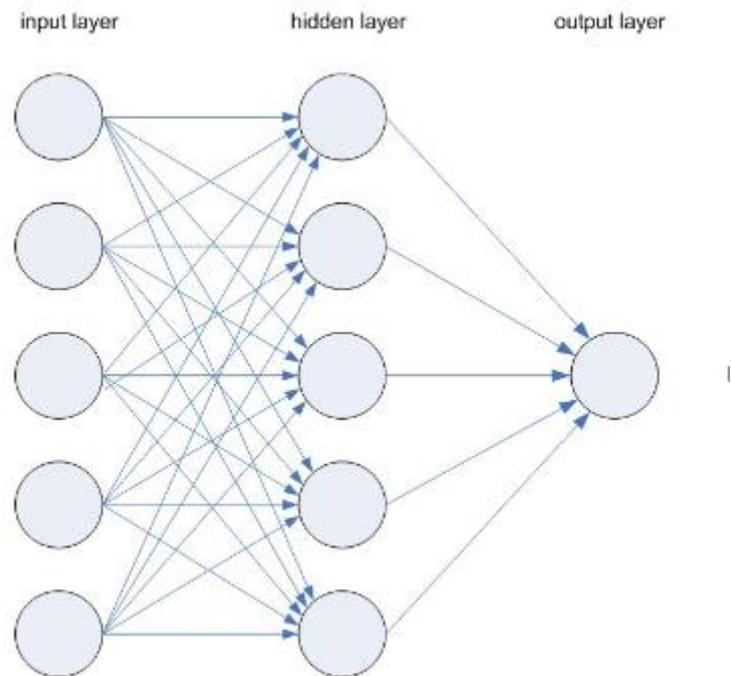


Figura 5. Esquema de una red neuronal. Extraído de [6]

Como podemos ver en la figura 5 las redes neuronales se organizan en capas.

- Capa de entrada: encargada de recibir los datos de entrada que recibirá la red neuronal.
- Capa de salida: presenta el resultado visible de la red, puede estar formada por una o varias unidades que representan el campo de destino.
- Capas ocultas: podrán ser una o varias, se denominan capas ocultas ya que se desconocen los valores de entrada y de salida de las mismas.

Las distintas neuronas se conectan entre sí con pesos variables. Los datos de entrada se propagan desde la primera capa y van pasando entre neuronas de las distintas capas, hasta llegar a la capa final.

A continuación, hablaremos de la forma de entrenar una red neuronal. En primer lugar, hemos de tener en cuenta que las redes neuronales son un tipo de aprendizaje automático en el que un ordenador desempeña una tarea mediante el análisis de ejemplos entrenados. Esto suele incluir una clasificación previa que implica etiquetar las imágenes con las que se va a entrenar el sistema. La red neuronal suele consistir de miles de neuronas conectadas entre sí que procesan la información de forma simple.

Normalmente la información sólo viaja en un sentido, desde la capa de entrada a la de salida.

Como podemos ver en la figura 5, una sola neurona puede estar conectada a varias neuronas de la capa anterior, de las que recibirá datos; y a varias neuronas de la capa posterior, a las que enviará datos. Cada una de las conexiones entre neuronas tendrá asignado un peso y cuando la red está activada, cada una de las neuronas recibirá uno dato distinto, que multiplicará por el valor del peso de su conexión. Finalmente sumará todos los productos resultantes y obtendrá un número como resultado. Si dicho resultado se encuentra por debajo de un valor umbral el nodo no enviará datos a la siguiente capa. En cambio, si el resultado excediera el umbral la neurona enviará la suma de todas sus entradas con el peso aplicado por todas sus conexiones de salida.

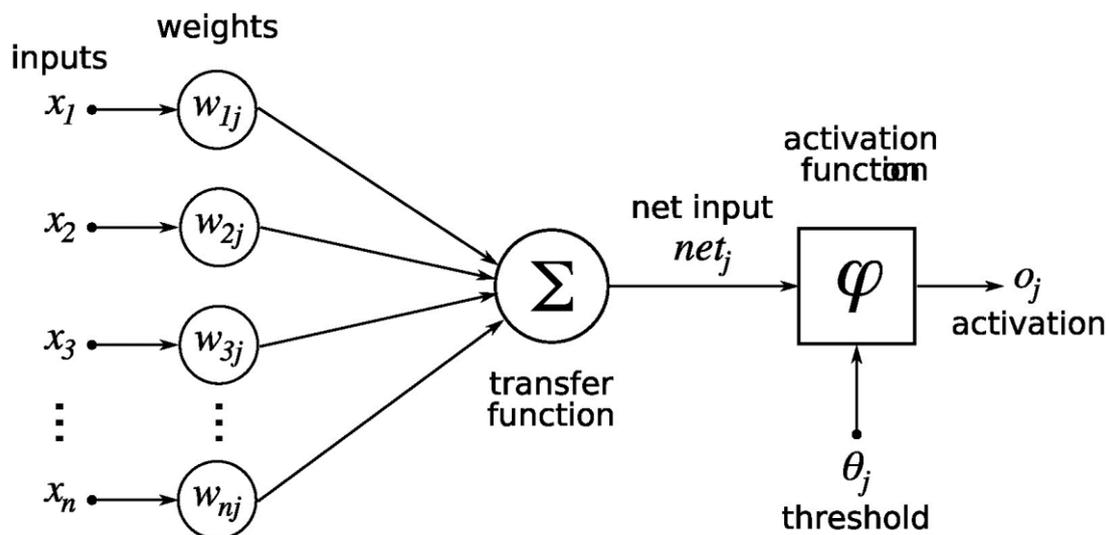


Figura 6. Explicación del proceso de entrenamiento. Extraído de [7]

En la figura 6 se muestra el proceso de entrenamiento explicado anteriormente. La función de activación mostrada en la figura es la responsable de evaluar si se activará o no la neurona, un proceso de activación implica el envío de datos, en función de si el valor "net input" es mayor o menor que el valor umbral "threshold".

Durante el proceso de entrenamiento de la red neuronal todos sus pesos y valores umbral son inicializados a valores aleatorios. Los datos con los que se

va a entrenar son enviados a la capa de entrada y pasan a las siguientes capas mediante operaciones de suma y producto complejas hasta que alcanzan la capa de salida. Durante el entrenamiento los pesos y los umbrales se ajustan continuamente, hasta que los datos de entrenamiento que están etiquetados de la misma forma obtienen salidas similares.

A continuación, explicaremos los principales tipos de redes neuronales.

- Redes de perceptrones multicapa (MLP)

Son un tipo de redes neuronales artificiales, en las cuales la información viaja de la capa de entrada a la capa de salida. Todas las capas están conectadas entre sí y podrá tener entre una y varias capas ocultas. Básicamente es el modelo explicado en la figura 5. Es el esquema de red neuronal más básico.

- Redes neuronales recurrentes (RNN)

Son un tipo de redes neuronales artificiales que llevan a cabo la introducción de datos de forma secuencial. Este tipo de redes fueron desarrolladas para solventar el problema de que los datos de entrada fueran introducidos de forma secuencial. Cada entrada de la red neuronal consiste en un nuevo dato y los anteriores y cada neurona guardará información acerca de los cálculos previos. En estas redes se comparten los parámetros de peso en cada capa, estos pesos siguen ajustándose de acuerdo con el algoritmo de retropropagación y pendiente de gradiente. Este tipo de redes neuronales se utilizan principalmente en el procesamiento del lenguaje natural (NLP), debido a que trabajan con datos de entrada con una longitud desconocida, como en el caso de una inteligencia artificial que comprenda el lenguaje natural. En la figura 7 se muestra un esquema de una red neuronal recurrente.

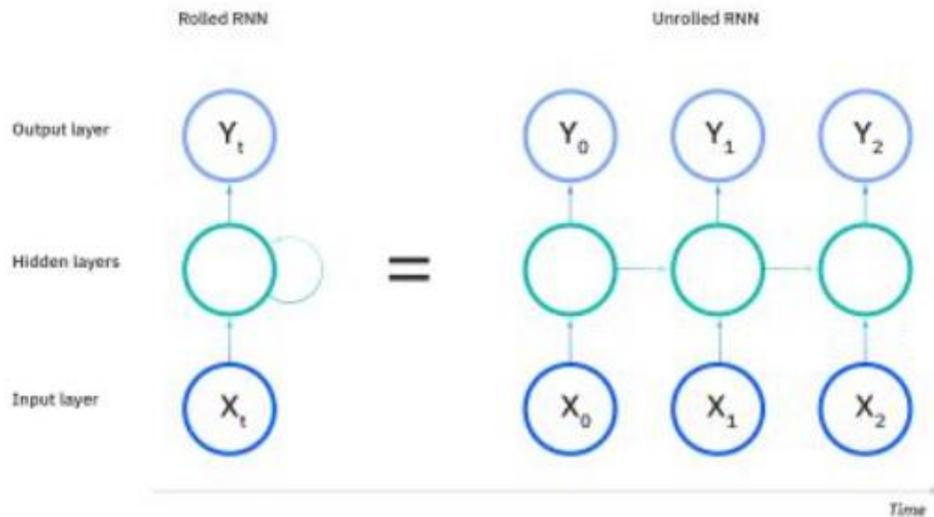


Figura 7. RNN. Extraído de [9]

- Redes neuronales convolucionales

Las redes neuronales convolucionales son un tipo de red neuronal que permite llevar a cabo tareas como clasificación de imágenes o reconocimiento de objetos de forma escalable mediante multiplicaciones de matrices para identificar patrones dentro de una imagen.

Una red neuronal computacional tiene tres tipos de capas:

- Capa convolucional
- Capa de agrupación (pooling)
- Capa clasificadora totalmente conectada (FC layer)

La capa convolucional es la primera capa de la red neuronal. En esta capa se lleva a cabo la mayor parte del coste computacional. Requiere varios componentes, que son los datos de entrada, un filtrado y un mapa de características. Si los datos de entrada son una imagen, esta estará compuesta por una matriz de píxeles en 3D, así que la entrada tendrá tres dimensiones, la altura, la anchura y la profundidad. También existirá un filtro que detectará si existe alguna característica. Este proceso completo se conoce como convolución. En la figura 8 se puede observar el proceso descrito.

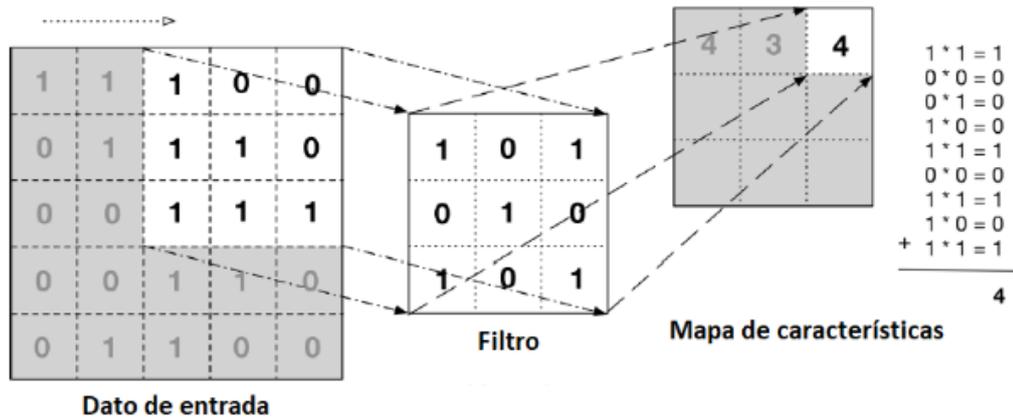


Figura 8. Operación de convolución. Extraído de [10]

Tras cada operación de convolución se aplica una unidad de rectificación lineal (ReLU), que supone una transformación del mapa de características para introducir no linealidad al modelo. En la figura 9 se describe el funcionamiento de una función ReLU, que sustituye los valores negativos por 0.

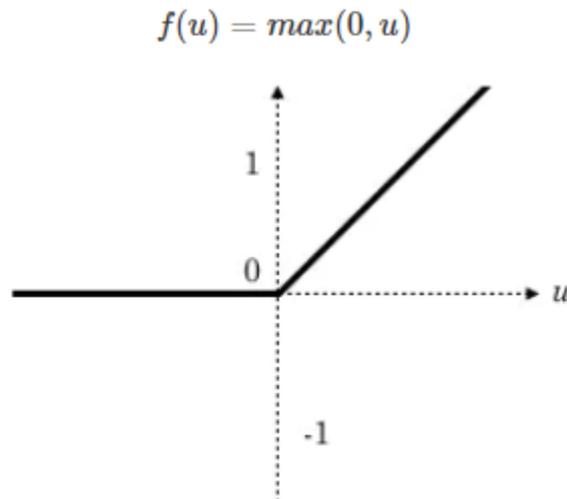


Figura 9. Función ReLU. Extraída de [10]

Tras la primera capa convolucional podríamos encontrar más capas de este tipo. De esta forma se conseguiría que la estructura de la CNN se vuelva jerárquica, ya que las capas posteriores verán los píxeles entre los campos receptivos de las capas anteriores.

La capa de agrupación o “pooling layer” reduce el número de parámetros de la entrada. Es similar a la operación convolucional, pero el filtro de esta operación

no contiene pesos, en su lugar, se aplica una función de agregación. Esta capa puede utilizar el “max pooling”, donde mientras el filtro selecciona dentro de los datos de entrada, solo selecciona el píxel con el máximo valor para enviar al vector de salida. A pesar de que este suele ser el método más usado, existe un segundo método, el de “average pooling”, que calcula el valor medio del campo medido y manda ese valor en vez del máximo al vector de salida.

Esta capa simplifica la información lo cual acaba resultando beneficioso ya que reduce problemas de complejidad y aumenta la eficiencia.

En la figura 10 se representa este proceso.

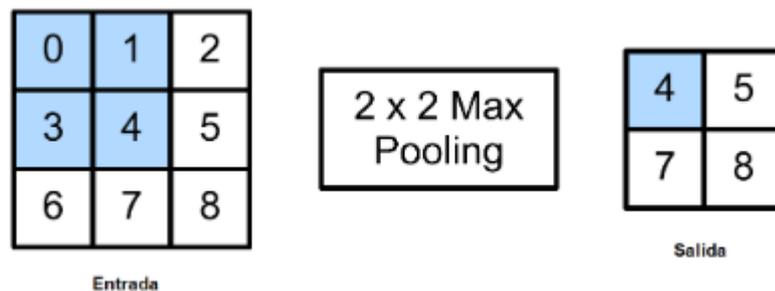


Figura 10. Capa de agrupación. Extraído de [10]

En último lugar, tenemos la capa clasificadora totalmente conectada. Esta capa conecta cada nodo de la capa de salida de forma directa a un nodo a la capa anterior. Permite llevar a cabo la clasificación basada en características extraídas en las capas previas con sus respectivos filtros. Normalmente, incluye una función de activación de tipo “softmax” para clasificar las entradas adecuadamente generando una probabilidad entre 0 y 1.

En la figura 11 se incluye un resumen de todo lo explicado en este apartado.

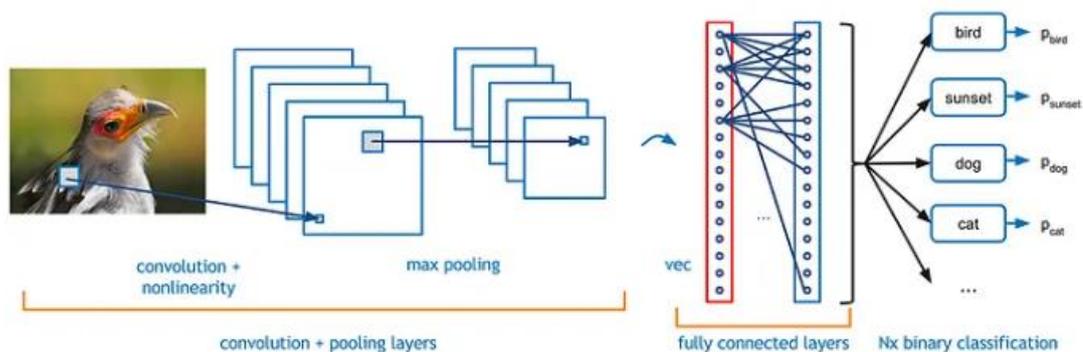


Figura 11. CNN. Extraído de [12]

Las principales arquitecturas de redes CNN son:

- AlexNet
- VGGNet
- GoogLeNet
- ResNet
- ZFNet

3.3 Text to speech (TTS)

El “text to speech” es una tecnología que lee el texto escrito en voz alta. Hoy en día, casi cualquier plataforma está preparada para que se pueda utilizar el TTS, la mayoría de los smartphones incluyen esta funcionalidad y la muchos de los programas, como Word o navegadores web también la incluyen.

El TTS puede crearse mediante la concatenación de sonidos grabados que se encuentran almacenados en una base de datos. La longitud de estos sonidos grabados puede variar, ya que cuanto más cortas sean las locuciones mayor variedad de salidas podrán ser generadas. Sin embargo, utilizando locuciones demasiado cortas se corre el riesgo de perder claridad. Un motor de TTS tiene dos partes principales. Mientras que tiene una parte front-end, que preprocesa el texto plano y después asigna transcripciones fonéticas a cada una de las palabras, el back-end se encargará de convertir estas representaciones fonéticas a voz.

El proceso se detallará a continuación a partir del diagrama mostrado en la figura 12.

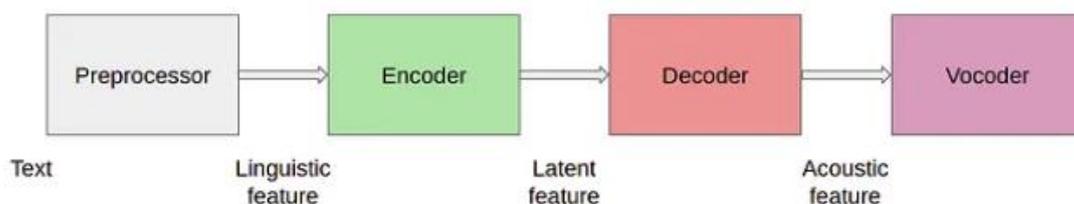


Figura 12. Estructura de un TTS. Extraído de [13]

La entrada al modelo es texto, que será transformado en audio a la salida del modelo. Pasará por una serie de transformaciones, de las cuales, la primera será el preprocesamiento del texto de entrada. El preprocesador llevará a cabo las siguientes tareas:

- Tokenización: se dividirán las frases en unidades de menor tamaño, normalmente palabras.
- Fonemas/ pronunciación: Se dividirá el texto en fonemas basados en su pronunciación.
- Duración de los fonemas: representa el tiempo total que durará cada fonema en el audio de salida.
- Tono de voz: es un factor clave para tener en cuenta ya que representa emociones a transmitir.
- Energía: indicará los coeficientes de frecuencias utilizados, lo que luego reflejará el volumen y la prosodia utilizados.

La salida lingüística solo contiene fonemas: la energía, el tono de voz y la duración se utilizan para entrenar predictores que consiguen que el modelo tenga una salida más natural.

El codificador sigue el diagrama presentado en la figura 13:

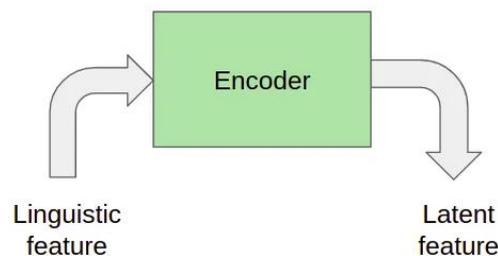


Figura 13. Codificador. Extraído de [13]

El codificador tiene como entrada fonemas y devuelve como salida un objeto de n-dimensiones. Este objeto que es luego la entrada del decodificador, se conoce como características latentes, las cuales también son usadas para la predicción del tono, la energía y la duración que controlan la naturalidad del audio.

El decodificador sigue el diagrama presentado en la figura 14:

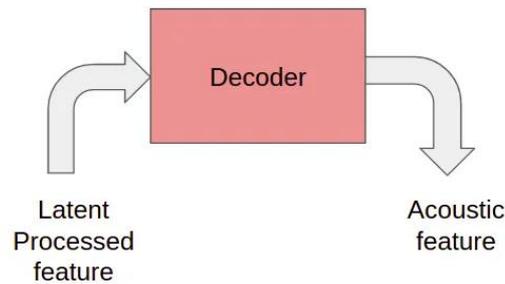


Figura 14. Decodificador. Extraído de [13]

El decodificador se usa para compartir las características latentes en características acústicas. La salida del decodificador se presenta en forma de espectrograma de Mel.

El siguiente paso es el codificador de voz o “vocoder”. Este dispositivo transforma las características acústicas (espectrograma de Mel) en ondas acústicas. Puede utilizar modelos matemáticos o redes neuronales para llevar a cabo ese proceso, aunque se suelen utilizar las segundas, ya que la eficiencia que se consigue es mayor.

El TTS tiene múltiples aplicaciones en la vida real, entre ellas cabe destacar:

- Lectura de contenidos para personas con discapacidad visual.
- Ayuda en las tareas de aprendizaje de nuevos idiomas.
- Mayor accesibilidad a contenido online.

4. Técnicas y herramientas

En este apartado se explicarán las técnicas y herramientas que se han utilizado en el desarrollo del proyecto. Las herramientas principales utilizadas durante el desarrollo del proyecto han sido: lenguajes de programación como JavaScript, Python, frameworks para el desarrollo web como Vue.js, herramienta Firebase de Google para el almacenamiento de los datos y herramientas auxiliares, como Microsoft Word entre otras.

Memoria. EyeSee

A continuación, se detallará el uso de cada una de estas herramientas para el desarrollo del proyecto.

4.1 Herramientas de desarrollo web

4.1.1 JavaScript

JavaScript es un lenguaje de programación ligero, interpretado o compilado just-in-time. Es conocido como un lenguaje de secuencias de comandos para webs y usado en entornos externos a navegadores, es un lenguaje orientado a objetos y basado en prototipos. Esto último implica que los objetos tendrán propiedades y métodos que se pueden compartir mediante la generalización y que pueden ser clonados y extendidos.

Se suele utilizar en el lado del cliente, debido a sus características dinámicas y su facilidad para implementar aplicaciones web.

4.1.2 HTML

HTML son las siglas de “HyperText Markup Language”. Este lenguaje es el estándar utilizado para documentos mostrados en la web y supone el componente más básico. Define el significado y la estructura del contenido de una web, utiliza enlaces para conectar varias páginas web entre sí. Estos vínculos pueden redirigir al usuario dentro del propio sitio web o a otros sitios web externos.

El lenguaje se basa en el uso de etiquetas para definir distintos tipos como imágenes, texto, etc. Las etiquetas se hayan delimitadas por los caracteres “<” y “>”. Este lenguaje no permite modificar libremente el estilo, por lo que se suele recurrir a otras herramientas como CSS, para modificar la apariencia de una web.

4.1.3 CSS

Este lenguaje toma su nombre de las siglas de “Cascading Style Sheets” y es el lenguaje de estilos utilizado para describir la presentación de documentos HTML.

Este lenguaje está basado en reglas formadas por las siguientes partes: un selector, el elemento HTML en el que comienza la regla, será el responsable de identificar a qué elementos dará estilo dicha regla. Detrás del selector encontramos la declaración, que especifica la propiedad del elemento a la que queremos dar estilo. Las propiedades son las diferentes formas en las que se puede modificar el estilo de un elemento y son asignadas un valor con el operador “:” que definirá la apariencia final del elemento. Las reglas se encapsulan entre llaves “{ }” y tras cada declaración dentro de una regla se usa el “;” como separador.

4.1.4 Vue.js

Vue.js es un framework de javascript utilizado en el proceso de construcción de interfaces de usuario. Utiliza como base HTML, CSS y JavaScript y proporciona un modelo de programación declarativo y basado en componentes.

Los componentes de Vue extienden los elementos de HTML con una sintaxis que permite describir de forma declarativa las salidas en HTML según estados de JavaScript.

Es un modelo reactivo, ya que automáticamente detecta los cambios de estado en el JavaScript y actualiza la DOM cuando se producen cambios.

Vue tiene en cuenta la diversidad de la web y se ha diseñado para ser flexible e incrementable. Los archivos de Vue.js tienen extensión “.vue” y usan un formato parecido al HTML, pero denominado Single-File Component (SFC), componentes de un solo archivo. Este tipo de archivos encapsulan la lógica en JavaScript, la plantilla en HTML y los estilos en CSS.

Además, Vue.js cuenta con una Options API que define la lógica de los componentes usando varias opciones: data, methods y mounted.

4.1.5 Vuetify

Vuetify es un framework de interfaz gráfica para Vue.js, que ofrece la posibilidad de mantener un estilo consistente en toda la aplicación, mientras ofrece suficientes posibilidades para distintas necesidades. Su diseño busca la sencillez y la facilidad de uso, ya que pretende ayudar a desarrolladores a diseñar interfaces de usuario de forma sencilla.

En este proyecto se ha utilizado Vuetify v3.

4.1.6 NPM

Las siglas NPM corresponden a “Node Package Library”. Esta herramienta consiste en una librería y registro para paquetes JavaScript. Además, cuenta con una línea de comandos, que permite instalación de paquetes y manejo de dependencias. Esta utilidad permite controlar los paquetes Node.js que están en uso. Node.js es un entorno de ejecución, que permite ejecutar JavaScript en el backend de una aplicación.

Para comenzar a utilizar el gestor de paquetes NPM, lo primero será ejecutar el comando “npm install”, lo cual creará todas las dependencias necesarias para el proyecto. En el archivo package.json se incluirán las versiones de las que depende el proyecto.

NPM incluye una serie de scripts, que permiten realizar acciones en el proyecto, entre ellos cabe destacar, npm test para ejecutar tests, npm build para compilar el proyecto y npm start para ejecutar el proyecto de forma local.

4.1.7 Firebase

Firebase es una plataforma de Google creada para facilitar el desarrollo de aplicaciones web y móvil priorizando la eficiencia, la rapidez y la seguridad.

Cuenta con tres tipos de módulos:

- Compilación: permiten iniciar el backend sin administrar servidores y escalar la aplicación de forma adaptada a su crecimiento. Dentro de este módulo encontramos, entre otros:
 - o Autenticación:

Memoria. EyeSee

- Base de datos en tiempo real
- Almacenamiento en la nube (Cloud Storage)
- Cloud Firestore
- Hosting
- Supervisión y lanzamiento: permiten gestionar el almacenamiento de la aplicación y comprobar su correcto funcionamiento. Encontramos, entre otros, dentro de este módulo:
 - Crashlytics
 - Performance monitoring
- Participación: módulo que busca como objetivo aumentar la participación de usuarios en la aplicación, mostrando información acerca del uso de la aplicación. Encontramos, entre otras, las siguientes herramientas:
 - Cloud Messaging
 - Analytics

4.2 Herramientas para el reconocimiento facial

4.2.1 Python

Python es un lenguaje de programación ampliamente utilizado en aplicaciones web, ciencia de datos, desarrollo software y tareas de aprendizaje automático. Los desarrolladores optan con frecuencia por este lenguaje de programación, ya que es eficiente y fácil de utilizar, además de portable.

Cuenta con una sintaxis sencilla, lo que facilita la lectura del código. Además, requiere menos líneas de código comparado con otros lenguajes de programación, existe código reutilizable para la gran mayoría de tareas y es un lenguaje muy sencillo de aprender.

Python es útil para escribir código backend, ya que cuenta con muchas bibliotecas para este propósito y también ofrece opciones de automatización para scripts. Uno de los grandes auges de este lenguaje de programación se ha producido gracias al desarrollo constante de tareas de aprendizaje automático, ya que este lenguaje es muy utilizado para este tipo de cuestiones.

4.2.2 Pytorch

Pytorch es una biblioteca de código abierto utilizada en aplicaciones aprendizaje automático especializada en reconocimiento de imágenes, procesamiento de lenguajes naturales (NLP) y deep learning.

Pytorch proporciona dos características a alto nivel:

- Computación de tensores con una aceleración fuerte mediante GPUs.
- Redes neuronales profundas basadas en un sistema de diferenciación automática de bases de datos.

Entre los principales beneficios del uso de Pytorch encontramos:

- La amplia comunidad que utiliza esta herramienta, con el soporte que eso supone.
- Integra librerías de Python como NumPy, SciPy, etc que son ampliamente utilizadas.
- Tiene soporte en la gran mayoría de plataformas en la nube.
- Soporta CPU, GPU y procesamiento paralelo, así como entrenamiento distribuido.

En la figura 15 podemos observar algunos de los principales beneficios de esta tecnología.

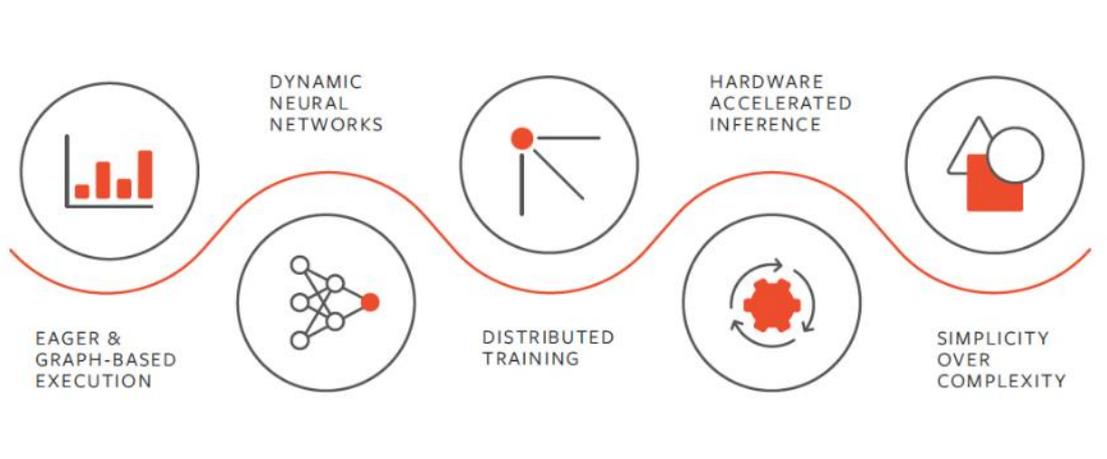


Figura 15. Pytorch. Extraído de [19]

4.3 Entorno de desarrollo

El entorno de desarrollo utilizado durante todo el proyecto ha sido Visual Studio Code. Esta herramienta permite crear aplicaciones con múltiples lenguajes de programación. Incluye opciones de autocompletado inteligente, depuración desde el propio editor y comandos git integrados. Además, ofrece una serie de extensiones personalizables que facilitan el trabajo del programador.

Para el desarrollo de la app se ha utilizado la versión v1.79.2 de mayo de 2023.

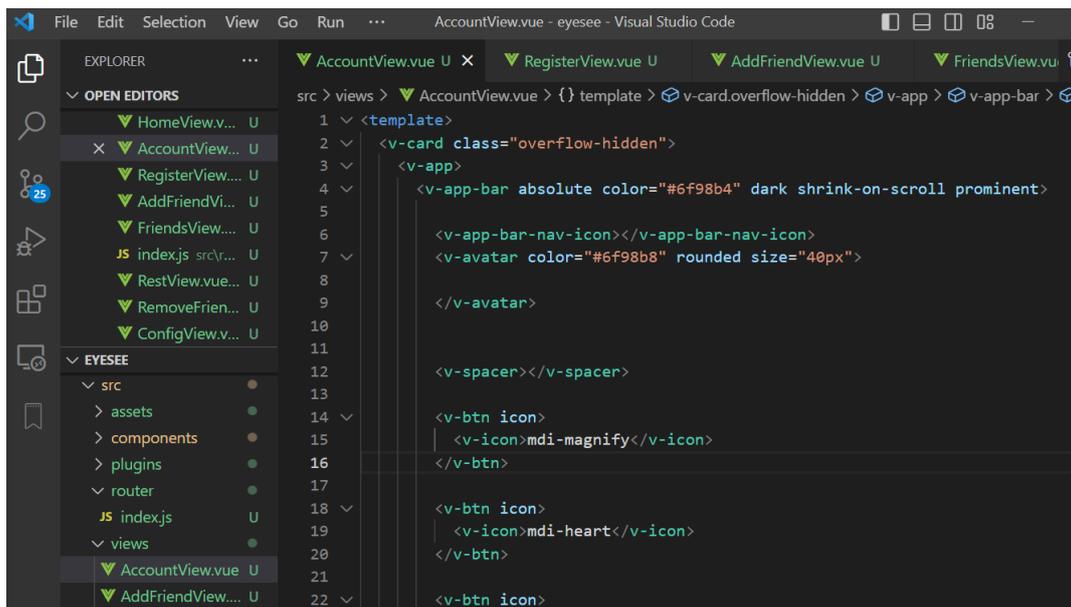


Figura 16. Visual Studio Code

4.4 Herramientas CASE

Las herramientas CASE son el conjunto de ayudas de ingeniería de software disponibles para facilitar el desarrollo de software, reduciendo el costo en términos de tiempo y dinero.

Estas herramientas tienen como objetivo mejorar la productividad y calidad del software, mejorar la planificación de un proyecto y ayudar a la reutilización del software, su portabilidad y la estandarización de la documentación.

4.4.1 Visual paradigm

Visual Paradigm es una suite completa de herramientas CASE, que utiliza el lenguaje unificado (UML). Proporciona soporte para modelado, generación de reportes y generación de código. También puede realizar ingeniería inversa de diagramas a código. Soporta 14 tipos de diagramas entre los cuales se encuentran: diagramas de componentes, de clases, de casos de uso, de comunicación, de máquinas de estado, de actividad...

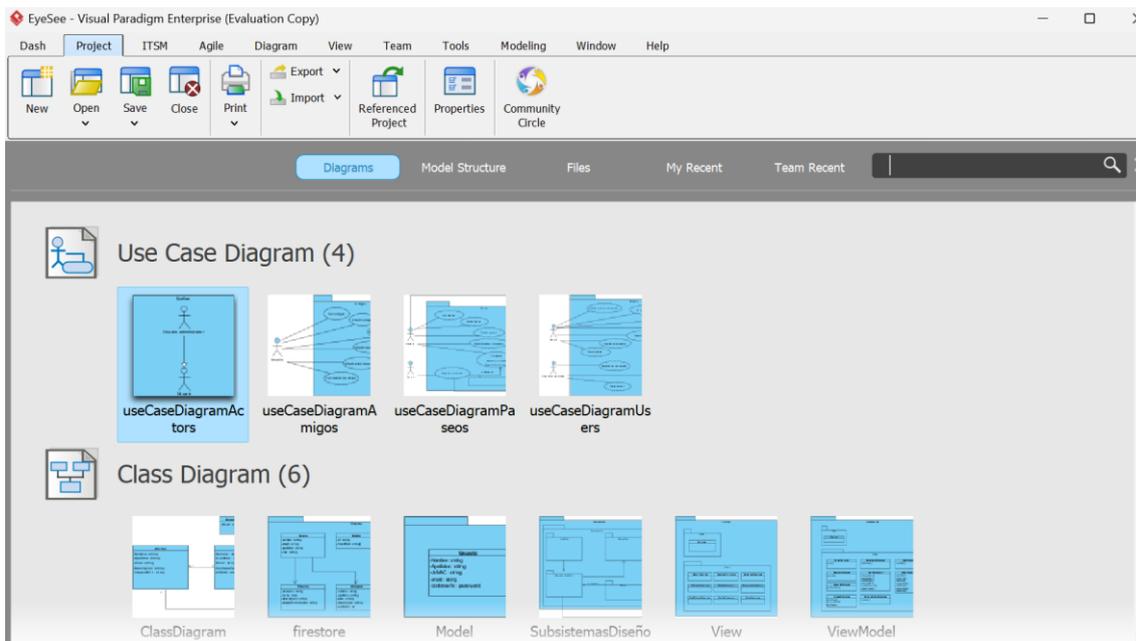


Figura 17. Visual Paradigm

En este proyecto se ha utilizado la versión 17.1

4.4.2 EZEstimate

Esta herramienta para gestión de proyectos se puede utilizar para estimar el esfuerzo que requerirá un proyecto determinado, mediante la metodología de Puntos de Casos de uso. Para ello se especificarán los casos de uso del proyecto, los actores y unos factores de complejidad técnicos y del entorno.

Memoria. EyeSee

Id	Module	Type	Name	complexity
1	Actores	Actor	Usuario no regi...	Simple
10	Usuarios	Usecase	Cambiar MAC	Simple
11	Usuarios	Usecase	Eliminar cuenta	Simple
12	Usuarios	Usecase	Cerrar sesion	Simple
13	Usuarios	Usecase	Restablecer co...	Simple
14	Amigos	Usecase	Ver amigos	Simple
15	Amigos	Usecase	Añadir amigo	Simple
16	Amigos	Usecase	Eliminar amigo	Simple
17	Amigos	Usecase	Añadir descripci...	Simple
18	Amigos	Usecase	Añadir alias ami...	Simple
19	Amigos	Usecase	Ver detalle amigo	Simple
2	Actores	Actor	Usuario	Simple
20	Paseos	Usecase	Ver paseos	Simple
21	Paseos	Usecase	Iniciar paseo	Simple
22	Paseos	Usecase	Finalizar paseo	Simple
23	Paseos	Usecase	Ver paseo	Simple
24	Paseos	Usecase	Eliminar paseo	Simple
25	Paseos	Usecase	Añadir detalles	Simple

Figura 18. EZEstimate

La versión que se ha utilizado es la 1.1.2.

4.4.3 REM

Las tablas de casos de uso se han realizado mediante la herramienta REM que ha sido desarrollada por la Universidad de Sevilla y que ofrece soporte para realizar las tablas de Durán y Bernárdez.

La versión utilizada es la 1.2.2.

4.4.4 Microsoft Project

Microsoft Project es una herramienta de gestión de proyectos, que permite realizar procesos de administración de forma colaborativa. Permite dar seguimiento a proyectos y detectar fallos en la planificación como la sobreasignación de recursos. También incluye la posibilidad de generar

Memoria. EyeSee

diagramas de Gantt, reportes y vistas personalizados que permiten gestionar de forma más eficiente el desarrollo de un proyecto.

Además, ofrece una visión de un cronograma en el que se recoge la disponibilidad de los miembros del equipo.

4.4.5 Microsoft Word

Es un programa de procesamiento de texto diseñado para la creación de documentos. Incluye funciones como la generación de índices o tablas de contenido, la inserción de gráficos... lo cual, trata de mejorar la eficacia en la generación de documentos.

Pertenece al paquete Microsoft Office y la versión utilizada es Microsoft Word 2019.

4.4.6 Adobe XD

Adobe XD es una herramienta de edición de gráficos que es utilizada para crear interfaces de páginas webs y aplicaciones. Permite realizar prototipos de interfaces en un espacio de tiempo muy corto, con múltiples opciones que simulan la navegación real del proyecto y que permiten detectar errores sin haber llegado al proceso de programación.

En este proyecto ha sido utilizado para crear el prototipo digital, en su versión 40.0.22.12.

5. Aspectos relevantes

En este apartado se expondrán los aspectos más relevantes en el desarrollo de este proyecto. Se expondrá un esquema general del sistema y posteriormente, se desarrollará con más detalle el proceso de creación de la aplicación.

5.1 Esquema general del sistema

La aplicación desarrollada en este Trabajo de Fin de Grado EyeSee es una aplicación web, en la cual los usuarios registrados pueden añadir a sus amigos o conocidos subiendo a la web una imagen de ellos. Los clientes podrán registrar paseos y luego guardar información relativa a estos, como su duración y los amigos encontrados.

Antes de comenzar el paseo, el usuario registrado comenzará a ejecutar un script de Python, que se mantendrá activo durante el tiempo que dure el paseo. Dicho script será el encargado de llevar a cabo el reconocimiento facial y si detectara, mediante una cámara que llevará el usuario, a una persona conocida, creará un aviso en la base de datos. La aplicación del cliente detectará la escritura del script en el sistema de almacenamiento y producirá dos avisos para el usuario: un cuadro de texto y una locución creada a partir del "text to speech". De este modo, el usuario sabrá qué persona conocida está aproximándose a él para saludarle.

5.2 Marco de trabajo

Durante el desarrollo del proyecto se ha utilizado el Proceso Unificado como marco de trabajo. Este proceso está dirigido por casos de uso y se pueden destacar las siguientes características:

- Conducido por casos de uso: los casos de uso se implementan para todas las funcionalidades del sistema. Esta es una forma de verificar y probar el mismo, los casos de uso incluyen una descripción de sus funciones y, por tanto, afectan a todas las fases y vistas.

Memoria. EyeSee

- Centrado en la arquitectura: la arquitectura se describirá mediante las diferentes vistas del sistema. En este proceso es importante establecer una arquitectura básica pronto, realizar prototipos y evaluaciones e ir refinando la propuesta a lo largo del curso del sistema.
- Iterativo e incremental: el proyecto se encuentra dividido en proyectos más pequeños, donde cada una de estas unidades constituye una iteración que acaba resultando en un incremento.

El Proceso Unificado se repite a lo largo de una serie de ciclos y cada uno de estos consta de cuatro fases:

- Inicio: se definirá el alcance del proyecto y se desarrollarán los casos de negocio.
- Elaboración: se planifica el proyecto, se especifican la mayoría de los casos de uso y se diseña la arquitectura del sistema.
- Construcción: se construye el producto.
- Transición: el producto se transforma en versión beta. Se corrigen problemas y se incorporan mejoras que haya surgido durante procesos de revisión.

Dentro de cada fase se descompone el trabajo en iteraciones incrementales. En estas iteraciones se llevarán a cabo tareas de las diferentes disciplinas del proceso unificado:

- Requisitos
- Análisis
- Diseño
- Implementación
- Pruebas

Al final de cada fase tendremos un hito, que supondrá la disponibilidad de un conjunto de componentes de software. Al final de cada hito, se tomarán decisiones respecto a la siguiente fase, se controlará el progreso del proyecto y se proporcionará la información necesaria para estimar la duración y los recursos de los siguientes proyectos. En la figura 19 se incluye el diagrama de ciclo de vida del proceso unificado.

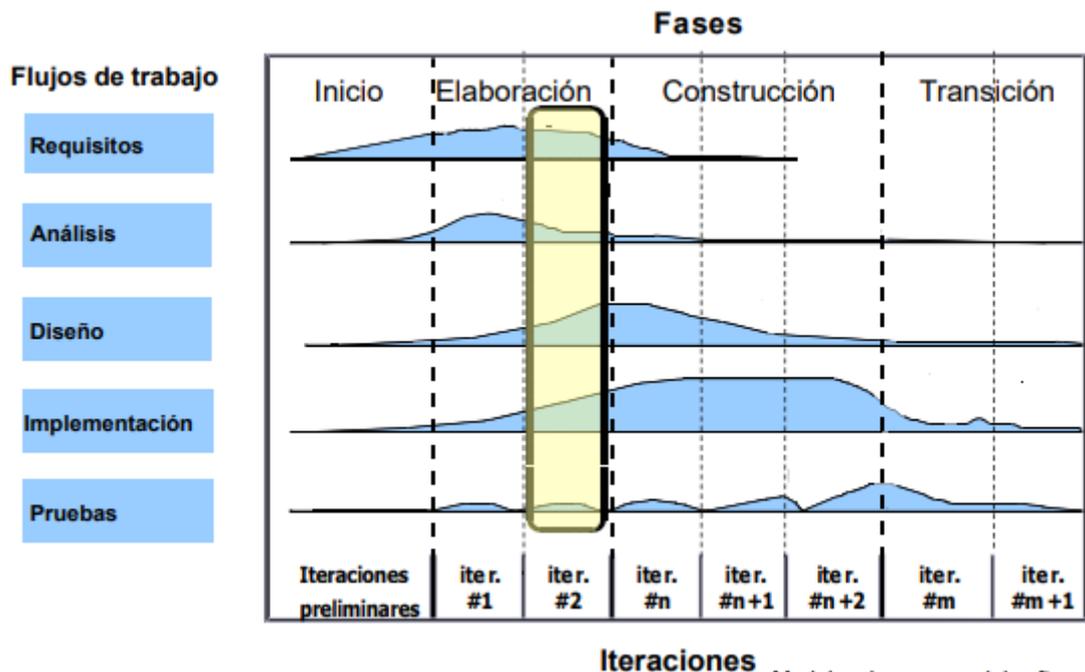


Figura 19. Ciclo de vida del proceso unificado. Extraído de [20]

A lo largo del proyecto se han realizado varias iteraciones. En el apartado “5.3 Planificación temporal”, se detallará la planificación de estas.

5.3 Estimación temporal

Esta tarea se ha llevado a cabo para predecir el esfuerzo del proyecto. Como cantidad de esfuerzo, tomaremos la relación entre tiempo y número de personas disponibles para realizar el proceso.

En nuestro caso realizaremos la estimación de esfuerzo del proyecto basado en el modelo de coste de puntos de caso de uso (UCP)¹.

Se ha utilizado la herramienta EZEestimate para la realización de los cálculos y se ha obtenido un esfuerzo total de 1001 horas por persona. En la figura 20 se pueden observar los resultados obtenidos.

The screenshot shows the EZEestimate software interface with the following data:

- Module:** Paseos
- Summary:** Total Modules: 4, Use cases: Simple 23, Average 0, Complex 0, Actors: Simple 3, Average 0, Complex 0.
- Estimation Summary:**
 - UAW: 3
 - UUCW: 115
 - UUPC = UAW + UUCW: 118
 - TFactor: 24
 - EFactor: 13
 - TCF = 0.6 + (.01*TFactor): 0.84
 - EF = 1.4 + (-0.03*EFactor): 1.01
 - UCP = UUPC*TCF*EF: 100.1112
 - Total Effort@ 10 Hrs/UCP: 1001.112
- Use case / Actor List:**

Id	Module	Type	Name	complexity
1	Actores	Actor	Usuario no regi...	Simple
10	Usuarios	Usecase	Cambiar MAC	Simple
11	Usuarios	Usecase	Eliminar cuenta	Simple
12	Usuarios	Usecase	Cerrar sesion	Simple
13	Usuarios	Usecase	Restablecer co...	Simple
14	Amigos	Usecase	Ver amigos	Simple
15	Amigos	Usecase	Añadir amigo	Simple
16	Amigos	Usecase	Eliminar amigo	Simple
17	Amigos	Usecase	Añadir descripci...	Simple
18	Amigos	Usecase	Añadir alias ami...	Simple
19	Amigos	Usecase	Ver detalle amigo	Simple
2	Actores	Actor	Usuario	Simple
20	Paseos	Usecase	Ver paseos	Simple
21	Paseos	Usecase	Iniciar paseo	Simple
22	Paseos	Usecase	Finalizar paseo	Simple
23	Paseos	Usecase	Ver paseo	Simple
24	Paseos	Usecase	Eliminar paseo	Simple
25	Paseos	Usecase	Añadir detalles	Simple

Figura 20. Estimación realizada con EZEestimate

5.4 Planificación temporal

Esta tarea ha sido de las primeras llevadas a cabo al comenzar el proyecto. La planificación temporal es un aspecto fundamental, ya que gracias a ella se puede conocer el tiempo y el esfuerzo necesario para llevar a cabo un proyecto, según el equipo de personas disponible para trabajar en él.

¹ Para información más detallada de la estimación del esfuerzo véase Anexo I. Planificación y estimación del esfuerzo.

Memoria. EyeSee

Este proceso se ha encuadrado dentro del marco de trabajo del “Proceso Unificado” y por ello, es necesario saber que las tareas estarán relacionadas entre ellas. Dichas dependencias influyen en la planificación temporal, ya que marcarán qué tareas se pueden realizar de forma simultánea y cuáles no².

A continuación, se mostrará la planificación temporal del proyecto por iteraciones obtenida mediante la herramienta Microsoft Project, así como el diagrama de Gantt del proyecto, con el camino crítico marcado en color rojo.

▸ Iteración 5	23 days	Fri 4/7/23	Wed 5/3/23	49	
▸ Requisitos	1 day	Fri 4/7/23	Fri 4/7/23		
Capturar requisitos residual	1 day	Fri 4/7/23	Fri 4/7/23		Silvia[50%]
▸ Análisis	1 day	Fri 4/7/23	Fri 4/7/23		
Revisar paquetes del análisis	1 day	Fri 4/7/23	Fri 4/7/23		Silvia[50%]
▸ Diseño	3 days	Sat 4/8/23	Tue 4/11/23		
Realizar modelo de arquitectura del diseño	3 days	Sat 4/8/23	Tue 4/11/23	66	Silvia
▸ Implementación	11 days	Wed 4/12/23	Mon 4/24/23		
Implementación del paquete Paseos	6 days	Wed 4/12/23	Tue 4/18/23	68	Silvia
Integración del paquete Amigos y el paquete	5 days	Wed 4/19/23	Mon 4/24/23	70	Silvia[50%]
▸ Pruebas	13 days	Wed 4/19/23	Wed 5/3/23		
Realizar pruebas del paquete Paseos	4 days	Wed 4/19/23	Sat 4/22/23	70	Silvia[50%]
Realizar pruebas de integración del paquete Amigos y el paquete	9 days	Mon 4/24/23	Wed 5/3/23	73	Silvia[50%]
Hito secundario. Fin de la iteración 5	0 days	Wed 5/3/23	Wed 5/3/23	62	

Figura 21. Planificación iteración 5

La planificación temporal completa del proyecto se puede ver en el Anexo I, junto con el diagrama de Gantt.

² Para conocer una versión más detallada de la planificación temporal véase el Anexo I. Planificación y estimación del esfuerzo.

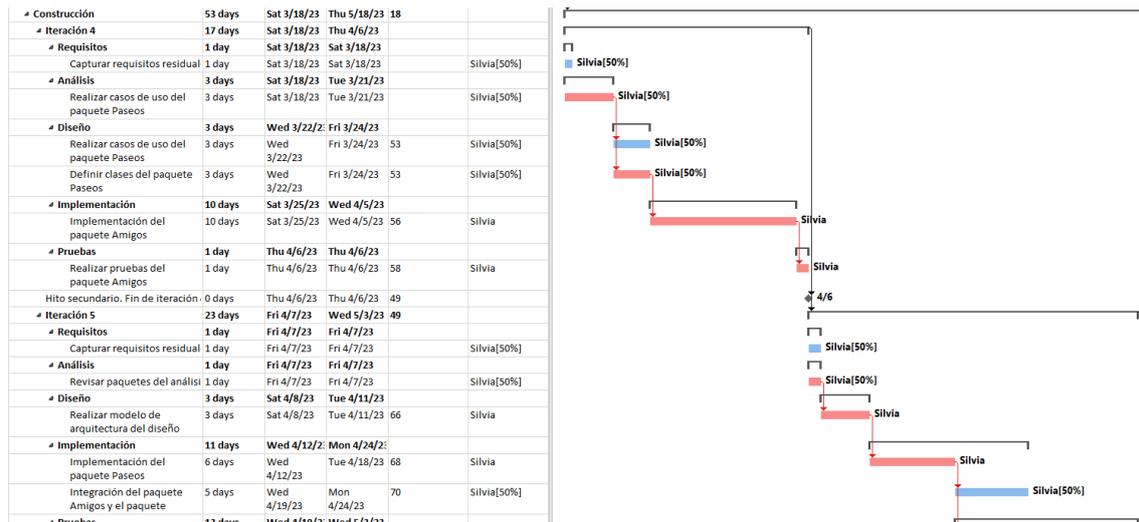


Figura 22. Diagrama de Gantt para la fase de construcción

5.5 Especificación de requisitos

La especificación de requisitos se lleva a cabo en la primera fase del proceso unificado. Incluye los siguientes apartados:

- Objetivos del sistema
- Modelo de requisitos del sistema
 - o Requisitos de almacenamiento de la información
 - o Requisitos no funcionales
 - o Definición de actores
 - o Definición de casos de uso
 - o Diagramas de casos de uso

A continuación, se mostrará el diagrama de casos de uso del sistema y del paquete paseos, realizado mediante la herramienta Visual Paradigm. Los diagramas de los demás paquetes se podrán consultar en el Anexo II.

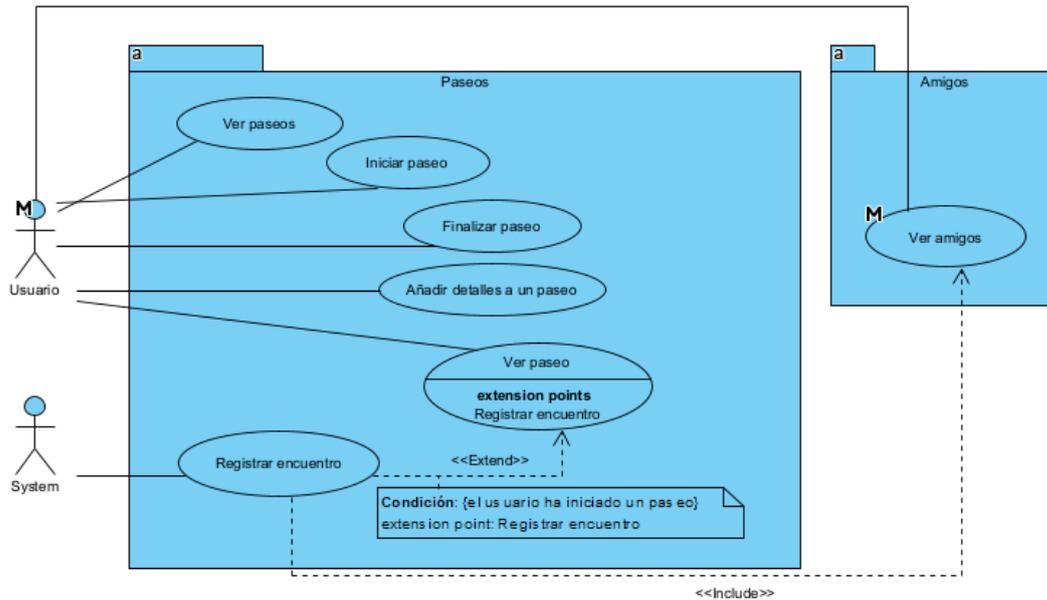


Figura 23. Diagrama de casos de uso del paquete paseos

En la figura 36 se muestra el diagrama de casos de uso del paquete paseos. En este paquete intervienen los actores: usuario, un usuario registrado, y el actor sistema. Los casos de uso más relevantes son “Iniciar paseo” y “Registrar encuentro”, que tiene como condición el inicio previo de un paseo.

En la figura 37 se muestra el diagrama de actores del sistema, donde existen tres tipos. En primer lugar, el usuario no registrado que es aquel que aún no ha iniciado sesión o no se ha registrado en el sistema. Por otro lado, existe un usuario administrador, que puede acceder a la información de todas las cuentas del sistema. Por último, tenemos el usuario que está ya registrado en el sistema y que ha iniciado sesión. Entre el administrador y el usuario existe una relación de generalización, ya que todas las acciones realizadas por el usuario pueden ser realizadas también por el usuario administrador.

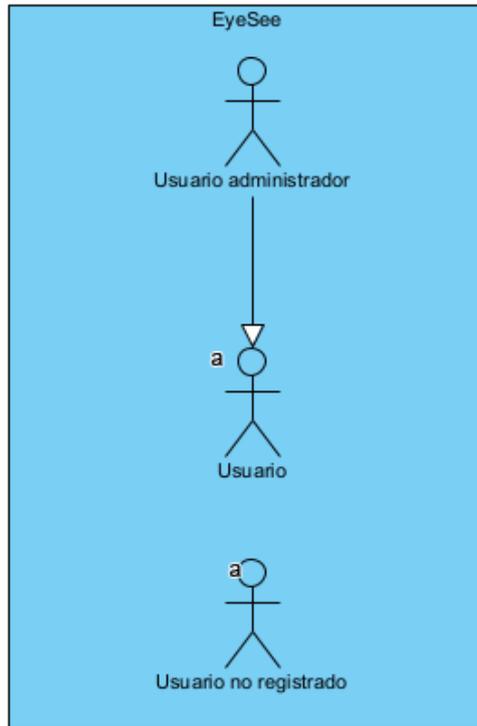


Figura 24. Diagrama de actores

A continuación, se expondrá un ejemplo de cada tipo de tabla, se siguió la metodología de Durán y Bernárdez, utilizada en el desarrollo de la especificación de requisitos.

Para obtener información más detallada se podrá consultar el Anexo II. Especificación de requisitos.

La tabla 1 pertenece al apartado “Participantes en el proyecto”.

Participante	Silvia Requejo López
Organización	Universidad de Salamanca
Rol	Alumna
Es desarrollador	Sí
Es cliente	No
Es usuario	No
Comentarios	Ninguno

Tabla 1. Participante Silvia Requejo López

5.5.1 Objetivos del sistema

Los objetivos del sistema que se definieron fueron:

- OBJ-0001 Gestión de usuarios
- OBJ-0002 Gestión de amigos
- OBJ-0003 Gestión de paseos
 - o OBJ-0004 Gestión de encuentros producidos
- OBJ-0005 Gestión de autenticación
- OBJ-0006 Gestión de reconocimiento facial

En la tabla 2 se presenta un ejemplo mediante el objetivo “Gestión de paseos”.

OBJ-0003	Gestión de paseos
Versión	1.0
Autores	- Silvia Requejo López
Fuentes	- André Sales Mendes - Gabriel Villarrubia González
Descripción	El sistema ha de permitir a los usuarios iniciar un paseo, finalizarlo, registrar los encuentros que se produzcan y editar o eliminar los datos de dicho paseo.
Subobjetivos	[OBJ-004] Gestión de encuentros producidos: el sistema ha de reconocer a los amigos de un usuario que éste se encuentre en un paseo. Ha de guardar dicha información y asociarla a cada paseo existente.
Importancia	Importante
Urgencia	Inmediatamente
Estado	Pendiente de verificación
Estabilidad	Media
Comentarios	Ninguno

Tabla 2. Gestión de paseos

5.5.2 Requisitos de información

Los requisitos de información que se definieron fueron:

- IRQ-0001 Información sobre usuarios
- IRQ-0002 Información sobre amigos
- IRQ-0003 Información sobre paseos
- IRQ-0004 Información sobre encuentros

En la tabla 3 se muestra un ejemplo de un requisito de información definido. Como podemos ver especifica la información que se va a almacenar relativa a usuarios dentro del sistema.

IRQ-0004	Información sobre encuentros
Versión	1.0
Autores	- Silvia Requejo López
Fuentes	- André Sales Mendes - Gabriel Villarrubia González
Dependencias	Ninguno
Descripción	El sistema deberá almacenar la información correspondiente a los encuentros producidos en paseos para cada usuario del sistema. En concreto se almacenarán los campos especificados en "Datos específicos".
Datos específicos	- Fecha - Hora - Amigo encontrado
Importancia	Importante
Urgencia	Inmediatamente
Estado	Pendiente de verificación
Estabilidad	Media
Comentarios	Ninguno

Tabla 3. Información sobre encuentros

5.5.3 Requisitos no funcionales

Se definió la siguiente lista de requisitos no funcionales:

- NFR-0001 Entorno de funcionamiento
- NFR-0002 Servicio web
- NFR-0003 Modularidad
- NFR-0004 Desacoplamiento
- NFR-0005 Rendimiento

Para consultar detalles acerca de cualquiera de estos requisitos, véase el Anexo II.

5.5.4 Actores

Se definieron los siguientes actores:

- ACT-0001 Usuario
- ACT-0002 Usuario no registrado
- ACT-0003 Usuario administrador

Mostraremos en la tabla 4 un ejemplo de la especificación del actor Usuario.

ACT-0001	Usuario
Versión	1.0
Autores	- Silvia Requejo López
Fuentes	- André Sales Mendes - Gabriel Villarrubia González
Descripción	Este actor representa un usuario con una sesión activa en el sistema.
Comentarios	Ninguno

Tabla 4. Actor usuario

5.5.5 Casos de uso

Los casos de uso definidos en el sistema se encuentran divididos según el paquete al que pertenecen.

- Paquete Usuarios
 - o UC0001 Registrarse

Memoria. EyeSee

- UC0002 Iniciar sesión
 - UC0003 Ver configuración
 - UC0004 Cambiar nombre de usuario
 - UC0005 Cambiar email
 - UC0006 Cambiar contraseña
 - UC0007 Cambiar MAC
 - UC0008 Eliminar cuenta
 - UC0009 Cerrar sesión
 - UC0010 Restablecer contraseña
- Paquete amigos
- UC0011 Ver amigos
 - UC0012 Añadir amigo
 - UC0013 Eliminar amigo
 - UC0014 Añadir descripción de amigo
 - UC0015 Añadir alias de amigo
 - UC0016 Ver detalle de amigo
- Paquete paseos
- UC0017 Ver paseos
 - UC0018 Iniciar paseo
 - UC0019 Finalizar paseo
 - UC0020 Ver paseo
 - UC0021 Eliminar paseo
 - UC0022 Añadir detalles a un paseo
 - UC0023 Registrar encuentro

A continuación, incluiremos una tabla de ejemplo de la definición de un caso de uso:

UC-0023	Registrar encuentro	
Versión	1.0	
Autores	- Silvia Requejo López	
Fuentes	- André Sales Mendes - Gabriel Villarrubia González	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se registre una coincidencia de la inteligencia artificial.	
Precondición	El actor Usuario registrado (ACT-0001) ha de haber iniciado un paseo.	
Secuencia normal	Paso	Acción
	1	El sistema registrará una coincidencia y la almacenará.
	2	El sistema registrará el cambio producido y mostrará los datos al usuario.
Postcondición		
Excepciones	Paso	Acción
Importancia	Importante	
Urgencia	Inmediatamente	
Estado	Pendiente de verificación	
Estabilidad	Media	
Comentarios	Ninguno	

Tabla 5. Registrar encuentro

5.5.6 Diagramas de casos de uso

En este apartado se realizaron los diagramas de casos de uso por paquetes de la aplicación. Los podemos observar en las figuras 34, 35 y 36.

5.6 Análisis de requisitos

En esta fase del proyecto se han analizado en profundidad todos los requisitos definidos en la fase anterior.

Para ello, se han se han seguido los siguientes pasos:

- Determinación del modelo del dominio.
- Realización de casos de uso del análisis.
- Elaboración de la propuesta de arquitectura del modelo del análisis.

Para obtener una descripción más extensa del sistema véase el Anexo III. Análisis de requisitos.

5.6.1 Modelo del dominio

El modelo del dominio ofrece una visión del sistema, incluyendo sus clases y las relaciones que podemos encontrar entre ellas.

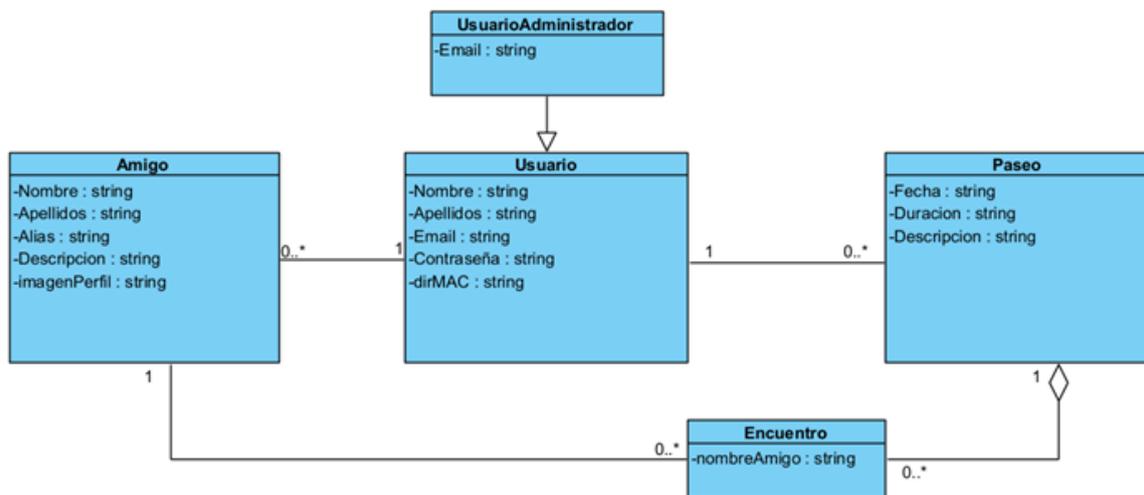


Figura 25. Modelo del dominio

Podemos ver que el sistema cuenta con una entidad usuario, que posee dos relaciones de asociación, una con la entidad amigo y otra con la entidad paseo. También podemos observar que la clase Usuario administrador hereda de la entidad usuario, teniendo en cuenta que en el sistema sólo podremos tener un único usuario administrador.

La entidad amigo posee dos relaciones de asociación, una con la entidad Usuario y otra con la entidad Encuentro, la cual posee una relación de agregación con la entidad paseo.

Por último, tenemos la entidad paseo que está relacionada tanto con encuentros como con usuarios de la forma ya descrita en el párrafo anterior.

5.6.2 Casos de uso del análisis

En esta sección se mostrará un diagrama de casos de uso del análisis, en concreto el diagrama para el caso de uso expuesto en la tabla 7: cambiar nombre de usuario.

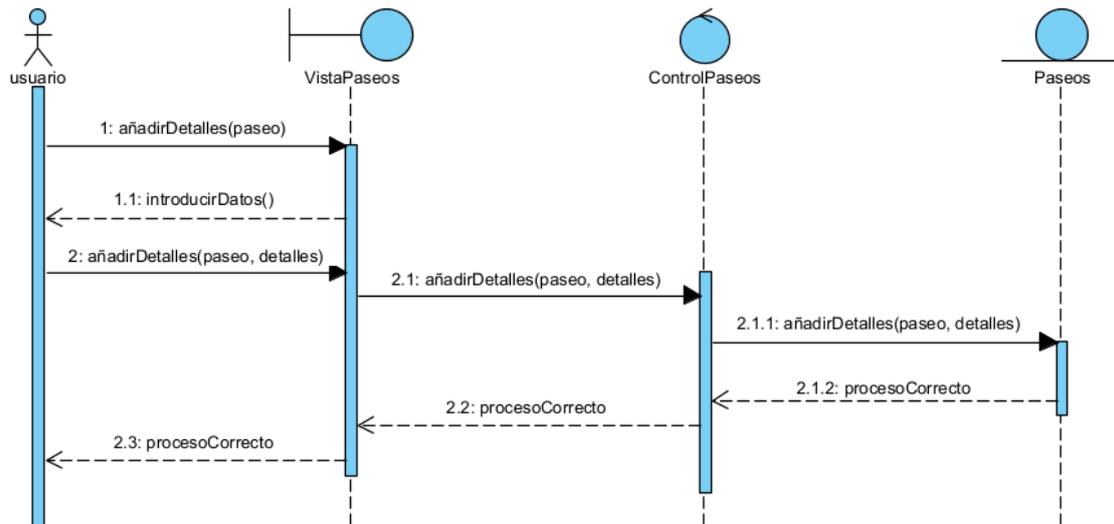


Figura 26. Añadir detalles a un paseo.

5.7 Diseño del sistema

En esta fase nos centraremos en el diseño del sistema, la cual es la primera fase del Proceso Unificado que pertenece al dominio de la solución.

Para obtener una versión más detallada de la etapa de diseño ha de consultarse el Anexo IV. Diseño del sistema.

Este apartado se encuentra estructurado de la siguiente forma:

- Modelo de Diseño
 - o Patrones arquitectónicos
 - o Subsistemas de diseño
 - o Clases de diseño
 - o Realización de casos de uso del diseño

- Diseño de la base de datos

5.7.1 Patrones arquitectónicos.

En el desarrollo de la aplicación web se ha utilizado el patrón Modelo-Vista-Modelo de vista, que es un patrón arquitectónico creado específicamente para desacoplar la interfaz de usuario de la lógica de negocio en el desarrollo de webs.

Consta de las siguientes partes:

- Vista: engloba los componentes web construidos
- Modelo: almacena los datos y el estado de la aplicación.
- Modelo de vista: implementa propiedades y comandos a los que la vista puede enlazar datos y notifica a la vista de cambios de estado mediante eventos de notificación. También aísla la vista del modelo y permite que el modelo evolucione independientemente de la vista.

Existen numerosas ventajas del uso de este patrón, entre ellas, la posibilidad de modificar la interfaz del usuario, sin que ello suponga cambios en el código del modelo o del modelo de vista. También permite a varios desarrolladores trabajar de forma simultánea en los diferentes componentes durante la implementación. En la figura 41 podemos observar el esquema del funcionamiento de este patrón.

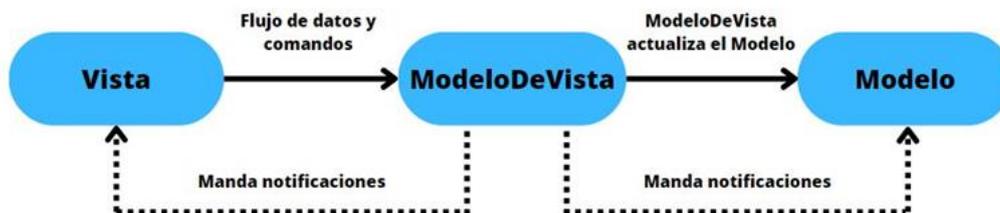


Figura 27. MVVM

El patrón Singleton también ha sido utilizado durante el desarrollo, ya que permite crear restricciones a la creación de objetos dentro de una clase. Esta es una forma de asegurar que las clases tengan una única instancia y que el acceso a la misma sea global.

En la figura 42 se muestra un esquema de este patrón.

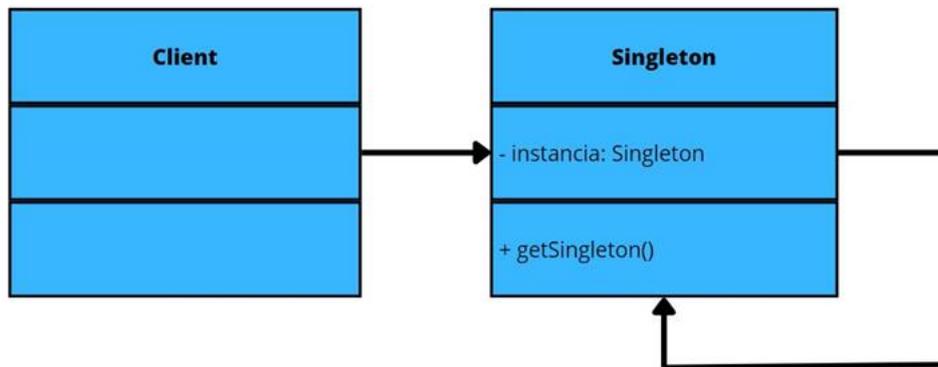


Figura 28. Patrón Singleton

5.7.2 Subsistemas de diseño

Los subsistemas de diseño definidos se muestran en la figura 43. Cabe destacar la existencia de tres paquetes distintos, el primero formado por el servicio web, el segundo formado por la API de firebase y el último formado por la API de reconocimiento facial

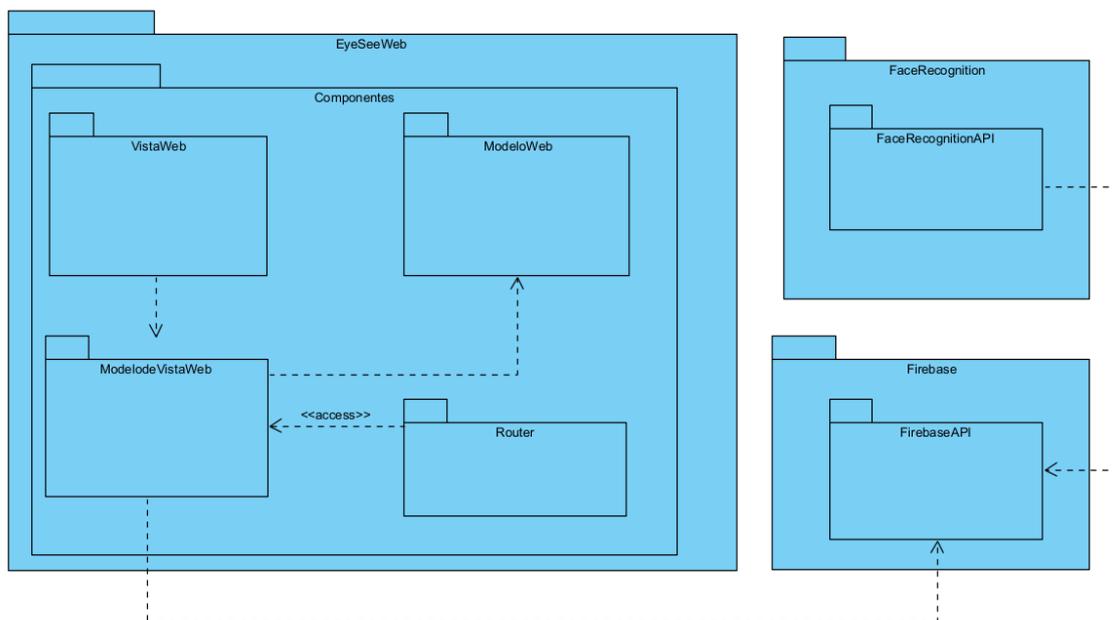


Figura 29. Subsistemas de diseño

5.7.3 Clases de diseño

En este apartado se especifican los contenidos del patrón Modelo-Vista-Modelo de vista, así como los métodos y atributos de cada paquete.

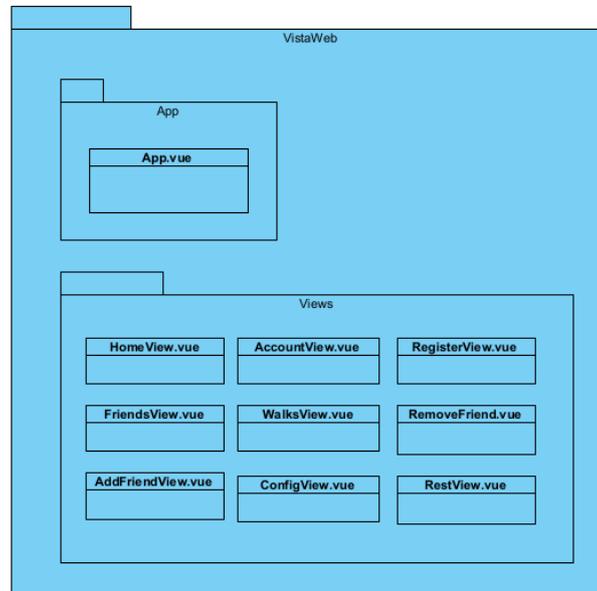


Figura 30. Paquete Vista

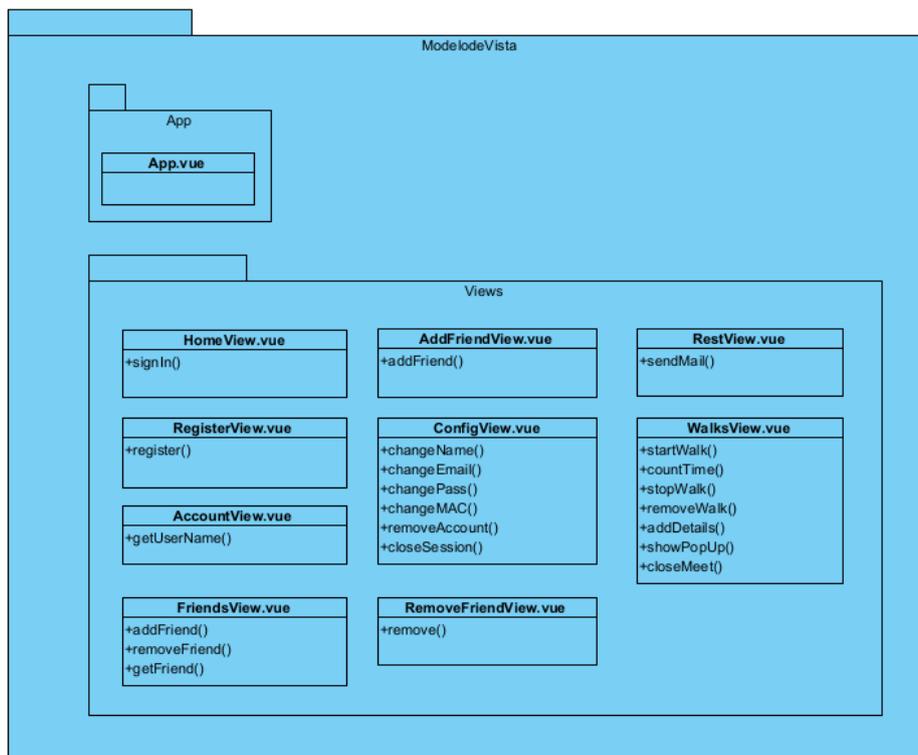


Figura 31. Paquete Modelo de vista

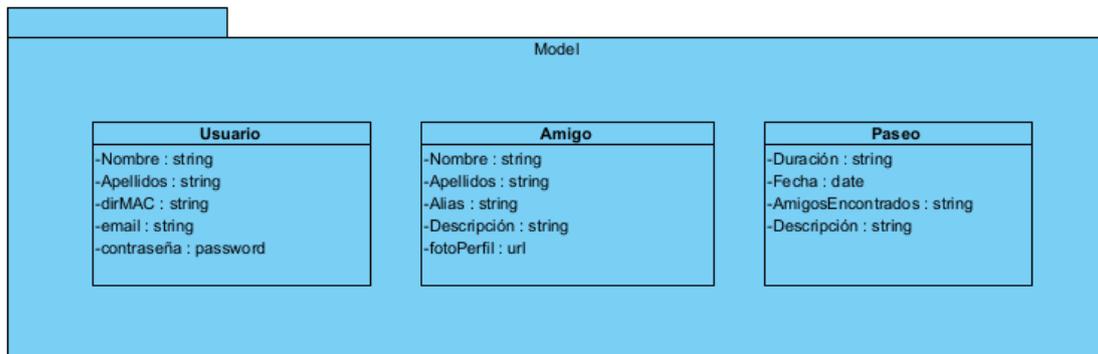


Figura 32. Paquete modelo

5.7.4 Realización de casos de uso de diseño

En este apartado se han realizado los diagramas de secuencia para los casos de uso del sistema. Se pondrá como ejemplo el diagrama de secuencia de la figura 47, que representa el caso de uso “Cambiar nombre usuario”.

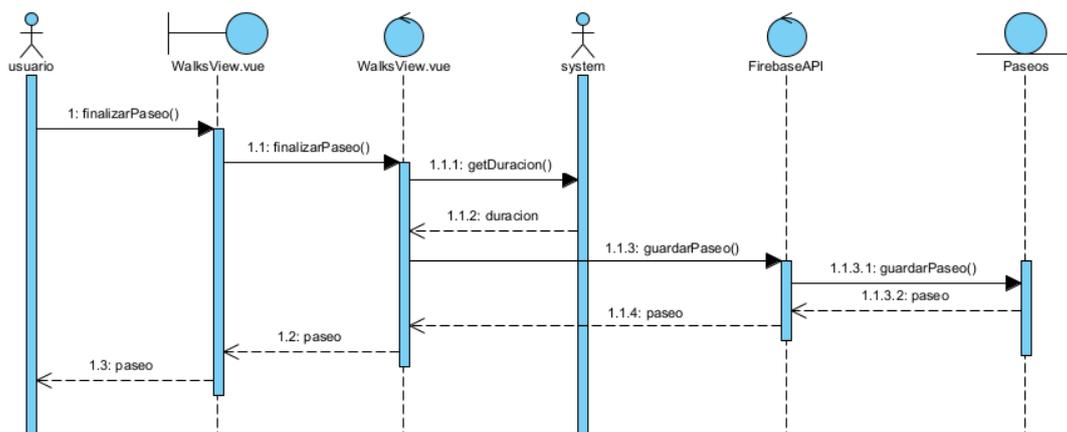


Figura 33. Diagrama de secuencia

5.7.5 Diseño de la base de datos

La base de datos de utilizada ha sido Firebase de Google. La estructura diseñada cuenta con dos colecciones de datos, la colección de usuarios y la colección de direcciones MAC. En la figura 48 se describe la lógica del sistema mediante un diagrama de clases.

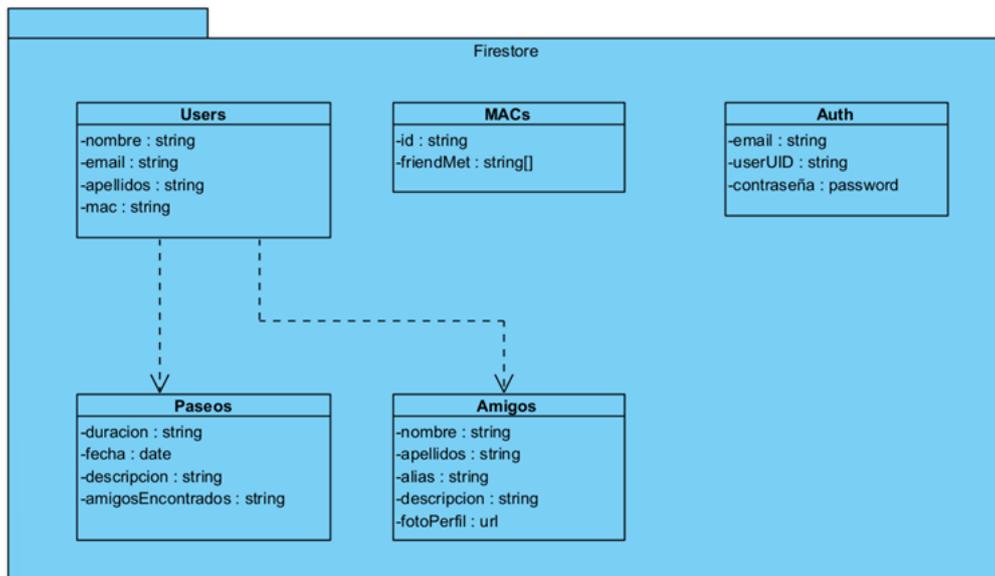


Figura 34. Diagrama de clases de Firestore

5.7.6 Implementación

Durante el proceso de implementación de la aplicación web, hemos utilizado las herramientas recogidas en el apartado 4, “Técnicas y herramientas”, tal y como se expone en dicho apartado hemos utilizado los lenguajes JavaScript, HTML y CSS mediante el framework de Vue.js, con Vuetify.

Al comienzo del desarrollo del sistema, ha sido necesario iniciarse en aspectos básicos de Vue y de Firebase, para ello se ha utilizado la documentación oficial de ambas herramientas. Durante la implementación, se ha utilizado la herramienta NPM, también mencionada en el apartado 4, para desplegar un servidor en local, mediante la orden “npm run serve”.

El primer paquete en ser implementado fue el paquete de gestión de usuarios. Para ello se estableció que, con cada registro de usuario, se almacenarían los datos de este en un documento, que tendría como nombre el identificador único asociado a dicho usuario (uid).

Para cada persona dada de alta en el sistema, se solicitaba una dirección de correo electrónico, el nombre y los apellidos y una contraseña, que tenía una longitud mínima especificada por razones de seguridad. El método de

autenticación utilizado fue el proporcionado por Firebase. También se utilizó el método de recuperación de contraseñas a los usuarios que proporciona el propio Firebase.

Posteriormente, en ese documento, con nombre del uid del usuario, se almacenarían todos los datos relativos al mismo. Al implementar el paquete amigos, se estableció que los amigos de un usuario se guardarían dentro de dicho documento, así como las urls de sus fotos de perfil.

Durante la implementación del paquete paseos, también se decidió almacenar la información relativa a los mismos en dicho documento.

Una vez se hubo implementado el paquete usuarios, se procedió a continuar con el de amigos. Durante la implementación de este paquete se produjeron mayores dificultades, especialmente con la lógica seguida para subir las imágenes a la base de datos. Finalmente, se decidió que los usuarios guardarían dentro de su documento las urls de las fotos de sus amigos. Dichas urls indicaban el lugar donde se almacenaban las imágenes dentro de la Firestore. Para cada amigo dado de alta en el sistema se requería un nombre, apellidos y una foto de perfil, además se podía proporcionar un alias y una descripción. Se añadió la opción de eliminar amigos existentes.

El último en implementarse fue el paquete paseos. En este paquete, se implementó la interfaz para los paseos, cada paseo se guardará asociado a un usuario en el documento del este. Se decidió que se recogería la duración de cada paseo, los amigos encontrados y se ofreció al usuario la opción de añadir una descripción a un paseo guardado.

5.7.7 Pruebas

En la fase de pruebas se han realizado numerosas comprobaciones acerca del funcionamiento del sistema, los fallos detectados se han ido corrigiendo con la mayor brevedad posible.

Tras la implementación de cada nueva funcionalidad, se han realizado pruebas de esta, controlando que el sistema se comportara como era deseado.

Finalmente, como se ha podido ver en la planificación, tras la implementación de un paquete completo se han llevado a cabo pruebas de su correcto funcionamiento. A su vez, se han realizado pruebas de funcionamiento entre paquetes tras la integración de cada uno de los mismos.

Por último, tras la implementación completa del sistema, se llevaron a cabo pruebas globales para detectar errores restantes.

5.8 Reconocimiento facial

El objetivo principal del sistema es la capacidad de llevar a cabo el reconocimiento facial. Esta tecnología fue desarrollada en un módulo externo y programada en Python, una vez se había comprobado su correcto funcionamiento, fue incorporada al sistema mediante una lógica que explicaremos en este apartado.

Se exploraron distintas opciones para el reconocimiento facial y se probaron diferentes tecnologías.

5.8.1 Investigación previa

Se investigaron diferentes opciones disponibles para obtener la tecnología de reconocimiento facial. Uno de los requisitos clave era encontrar un proyecto de código abierto. Se exploraron por tanto las distintas posibilidades y se compararon los valores de exactitud para cada una de ellas.

- Deepface: sistema de reconocimiento y análisis de atributos faciales, como edad, género, emoción... Une lo mejor de varios modelos, como el FaceNet de Google o el ArcFace. Cuenta con una exactitud del 97.5% en la tarea de reconocimiento facial y está desarrollada para Python.
- CompreFace: está basado en Docker y puede ser utilizado como un único servidor, o lanzado desde la nube. Cuenta con una API REST, que se puede utilizar para tareas de verificación facial, reconocimiento y detección faciales, entre otras. Soporta tanto modelos de CPU como de GPU y es fácilmente escalable. Cuenta con una exactitud del 99.6%.

Memoria. EyeSee

- Face Recognition: usa una API de Python y se basa en el uso de pytorch. Fue la tecnología escogida debido a su sencillez de uso y de adaptación. Cuenta con una exactitud del 99.3%
- InsightFace: librería de Python para análisis facial en 2D y 3D, que implementa numerosos algoritmos de reconocimiento, detección y alineamiento facial optimizados. Tiene una exactitud del 99.83%.
- FaceNet: librería de Python para el reconocimiento facial, que usa TensorFlow y cuenta con una exactitud del 99.6%.

En este proyecto nos decantamos por la librería face-recognition, ya que resultó sencilla de adaptar para el propósito de la aplicación, fue fácil de utilizar desde el primer momento y, aunque el lenguaje de programación no era conocido, a la hora de empezar a desarrollar con leer algunos tutoriales fue suficiente para comprender su funcionamiento.

El script fue creado para leer las imágenes de un directorio local, y las primeras pruebas se llevaron a cabo de esa forma. Una vez se comprobó que el script funcionaba, se esperó a que estuviera implementada la aplicación.

Cuando la app se encontraba implementada, se comenzó a usar el script de reconocimiento facial. Se adaptó de forma que al ejecutarse descargara las imágenes de los amigos asociados al usuario que tenía la aplicación. Es necesario que un usuario introduzca la dirección MAC de su dispositivo antes de empezar un paseo. El miniordenador donde se ejecuta el script tiene una dirección MAC, que es detectada por el programa de reconocimiento facial, el usuario también conocerá dicha dirección y la introducirá en la aplicación.

Cuando el script se comienza a ejecutar, busca el usuario que tenga asociada la dirección MAC que tiene el miniordenador y se descarga en local los datos de los amigos de dicho usuario. Creará también un nuevo documento en la colección "MACs" de firebase, donde guardará los encuentros que se hayan producido.

La aplicación, mientras detecte que hay un paseo en curso, escuchará los cambios que se produzcan en dicha colección y, si se escribe el nombre de un amigo, lo mostrará por pantalla a la vez que dirá el nombre de este en voz alta con una locución como la que sigue: “Se va a encontrar a (nombre del amigo)”.

Esta última parte de integración requirió juntar la aplicación web con el script de Python y resultó más compleja, ya que hubo que ejecutar cambios en la lógica del sistema.

5.9 Diseño de la aplicación

La aplicación está pensada para un público que posee un cierto grado de discapacidad visual, lo cual ha afectado a las decisiones tomadas respecto al diseño. En primer lugar, se han intentado utilizar colores de fondo sobre los cuales resulte fácil distinguir el texto. En concreto, se han utilizado fondos blancos con motivos en azul oscuro y se han usado caracteres en negrita para destacar la información.

Se ha buscado un diseño sencillo y modular, que distinga bien las diferentes partes de la web. Los amigos, por ejemplo, se muestran en distintos apartados claramente diferenciados y con la imagen de cada amigo en primer plano, para que sea fácilmente reconocible.

La interfaz de paseos se ha diseñado de forma minimalista, se pueden ver todos los paseos con su fecha correspondiente desde la página de inicio y si se desean consultar detalles acerca de los mismos sólo es necesario pulsar sobre ellos y se desplegará la información relativa a cada uno.

Finalmente se tiene en cuenta que los móviles de personas con algún grado de discapacidad visual suelen estar configurados con ayudas como mayor tamaño de texto o incluso “text to speech”, ayudas de las cuales se beneficiará nuestra app. Sin embargo, se cuenta con la posibilidad de que no dispongan de dichas ayudas y, en ese caso, un familiar o una persona cercana podría ayudarles en el proceso de configuración de la aplicación.

Memoria. EyeSee

A la hora de iniciar un paseo, ya no sería necesaria la ayuda ya que los paseos tienen incorporada la funcionalidad del “text to speech” y por tanto no necesitarán la ayuda de otras personas para utilizar la aplicación.

A continuación, se mostrarán las capturas de pantalla extraídas de la aplicación.

The image displays two side-by-side screenshots of a mobile application's registration screen. Both screens feature a white background with a light blue border. The left screenshot is titled "Registrarse" and includes the "EyeSee" logo, which is a stylized blue eye. Below the logo, there are five input fields: "Nombre" (Silvia), "Apellidos" (Requejo), "Email" (silviareqlop@usal.es), "Contraseña" (masked with dots), and "Repetir contraseña" (masked with dots). At the bottom, there is a checkbox for "Estoy de acuerdo con los términos y condiciones". The right screenshot features a large blue eye icon at the top. Below it, the same five input fields are present, but the "Repetir contraseña" field is empty. At the bottom right, there is a blue button labeled "FINALIZAR >".

Figura 35. Registro

La figura 49 muestra el formulario de registro y los datos solicitados al usuario para crear una cuenta en el sistema. En la figura 50 podemos ver la página principal de la web. Desde esta página se puede acceder al registro, o solicitar una recuperación de la contraseña. En la figura 51 se muestra el proceso de recuperación de la contraseña, que utiliza una funcionalidad específica de Firebase.



Figura 36. Inicio de sesión

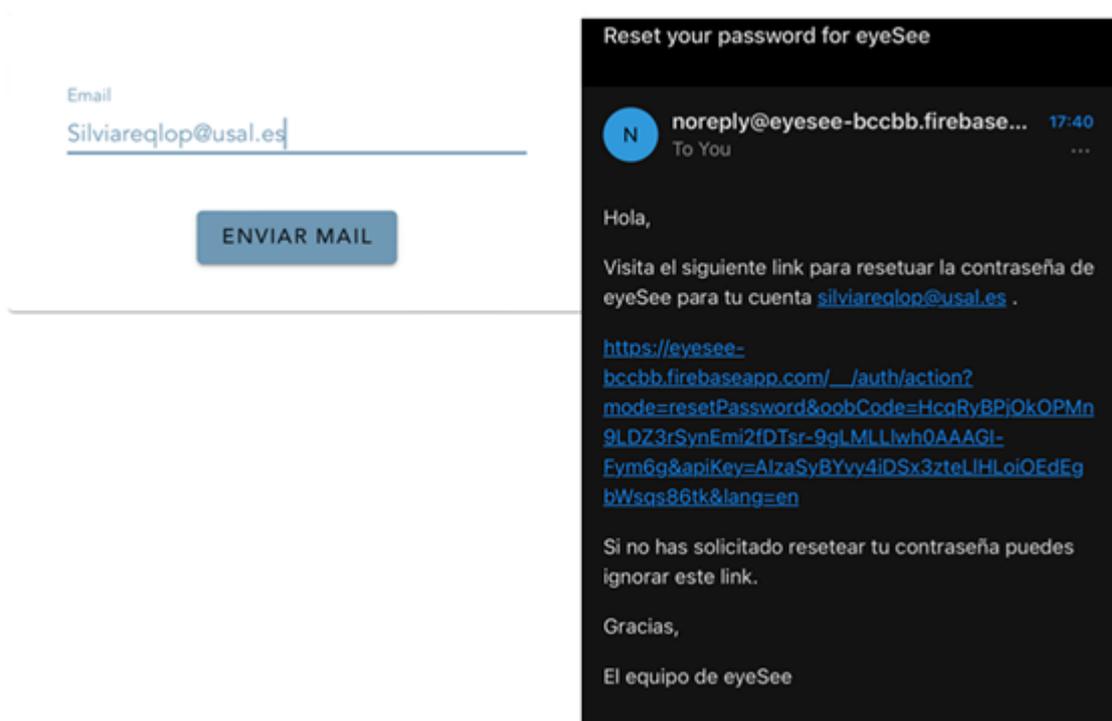


Figura 37. Recuperación de la contraseña

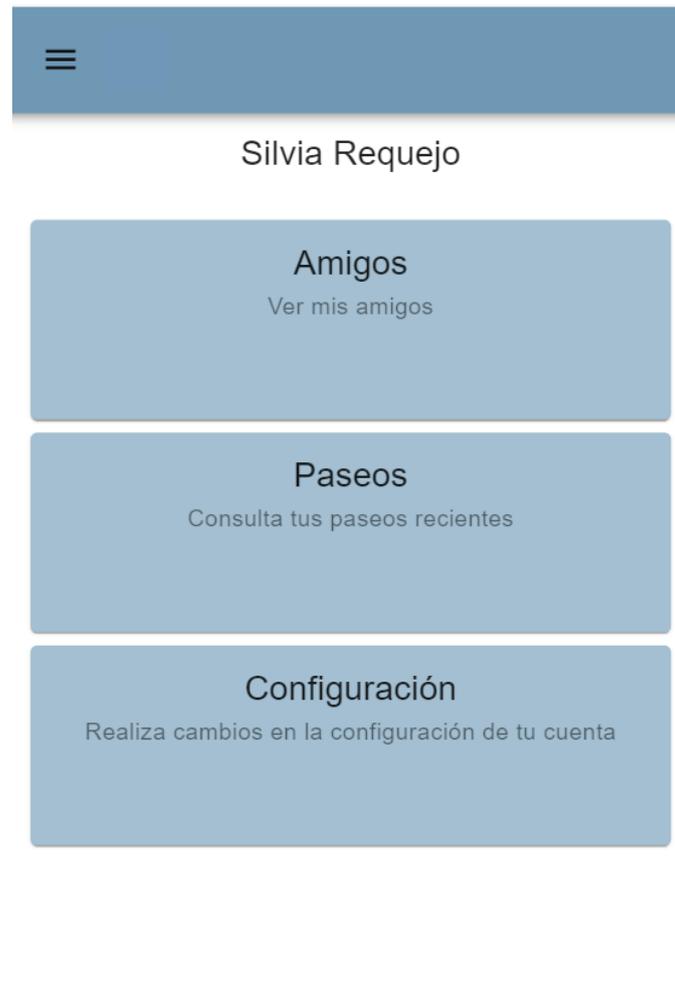


Figura 38. Pantalla de inicio

En la figura 52 se puede ver la pantalla de inicio para un usuario registrado en el sistema. Tiene tres opciones principales, que son “Amigos”, “Paseos” y “Configuración”.

Desde esta pantalla se puede acceder a todas las funcionalidades del sistema. Por otro lado, desde mis amigos se pueden editar amigos en el desplegable que tenemos al comienzo de la pantalla. También se podrán editar desde el menú que aparece de la parte izquierda de la pantalla. Además, deslizando hacia abajo desde esta pantalla se muestra la lista de amigos añadidos.

Los tres iconos de la parte derecha de la barra de navegación están presentes en todos los submenús y permiten acceder a amigos, paseos y configuración respectivamente. En la figura 54 se puede ver el proceso de añadir un amigo

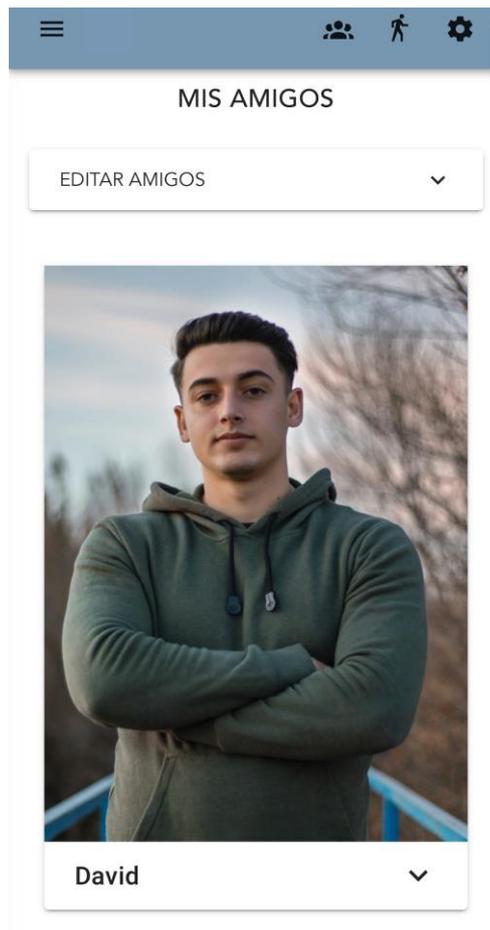


Figura 39. Pantalla mis amigos

Nombre
David

Apellidos
González

Alias
David

Descripción
David amigo biblioteca

ELIJA UNA IMAGEN

File input
IMG_6789.jpeg

ACEPTAR

Figura 40. Añadir amigo



Figura 41. Eliminar amigo

En la figura 55 se puede ver el proceso de eliminación de un amigo.

En las siguientes figuras se mostrarán las interfaces de la sección “paseos”.



Figura 42. Página principal paseos.

Memoria. EyeSee

Desde la barra de navegación se puede acceder a los otros submenús igual que en el caso anterior. El botón de iniciar paseo comenzará a contar el tiempo en el banner que se encuentra en la parte superior.

Al pulsar sobre un paseo, se pueden observar los detalles de este. El proceso de añadir descripción a un paseo se muestra en la figura 57.

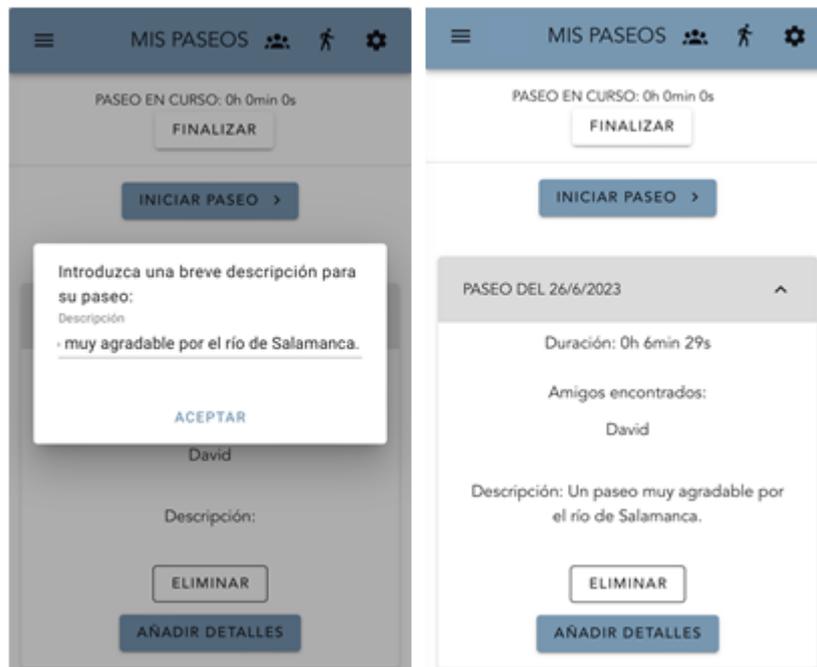


Figura 43. Añadir descripción.

El botón eliminar permite borrar de forma permanente un paseo.

Si durante un paseo se produce un encuentro se mostrará un banner como el de la figura 58.



Figura 44. Encuentro producido

En la figura 58 se puede observar la página principal de configuración de la web. En ella se pueden modificar los datos personales del usuario.



Figura 45. Pantalla de configuración.

The screenshot displays the 'CONFIGURACIÓN' (Settings) interface. At the top, there is a blue header bar with a menu icon on the left and user, location, and settings icons on the right. Below the header, the title 'CONFIGURACIÓN' is centered. The settings are organized into several sections:

- Personal Information:** A white box containing 'Nombre' (Name) set to 'Silvia Requejo' and 'Email' set to 'silviusass22@gmail.com', both with dropdown arrows.
- Password Section:** A grey header 'Contraseña' with an upward arrow. Below it are two input fields: 'Introduzca la nueva contraseña' and 'Confirme la nueva contraseña'. A blue button labeled 'CAMBIAR CONTRASEÑA' is positioned below these fields.
- Device Information:** A white box containing 'MAC del dispositivo' (Device MAC) set to 'BC:54:2F:CF:63:7C' with a dropdown arrow.
- Action Buttons:** Two blue buttons are located at the bottom: 'CERRAR SESIÓN' (Log Out) and 'ELIMINAR CUENTA' (Delete Account).

Figura 46. Modificación de contraseña.

Para más detalle en este apartado se podrá consultar el Anexo VI. Manual de usuario.

6. Conclusiones

El proyecto Trabajo de Fin de Grado ha finalizado habiendo cumplido satisfactoriamente los objetivos fijados al inicio del mismo, que eran la gestión de usuarios, paquete encargado del registro de usuarios en el sistema, así como de gestionar la información de los mismos; la gestión de amigos, paquete encargado de almacenar y realizar cambios en la información de los amigos guardados para cada usuario; la gestión de paseos, encargado de tramitar los aspectos relacionados con los paseos en el sistema; la gestión de autenticación, encargado de la alta y sesión activa de los usuarios y la gestión de reconocimiento facial, encargado de tramitar los encuentros producidos y gestionar las alertas de los mismos.

Este proyecto ha supuesto, sin duda, un reto para mí, ya que suponía trabajar con herramientas que desconocía, como la tecnología utilizada para el reconocimiento facial, mientras usaba lenguajes de programación como JavaScript, Python, con los que no me encontraba familiarizada al principio del proyecto. Sin embargo, hoy en día puedo afirmar que, gracias al trabajo realizado, domino dichos lenguajes de programación. También he aprendido a usar un framework para el desarrollo web, como Vue.js que es ampliamente utilizado y facilita las tareas de programación.

Este proyecto también ha sido un reto por otros motivos, ya que como ya he mencionado existía una implicación personal para mí en el proyecto. La motivación surgió, ya que al observar a personas de mi entorno descubrí las dificultades que pueden surgir, al padecer algún grado de discapacidad visual, en su vida cotidiana. Al finalizar el proyecto puedo decir que ha sido un proyecto exitoso, ya que ha conseguido el objetivo de ayudar a dichas personas y facilitarles aspectos sencillos pero relevantes en su día a día.

Este Trabajo de Fin de Grado ha sido una muy buena oportunidad para poder poner en práctica todos los conocimientos adquiridos en todas las asignaturas de Ingeniería del Software cursadas en la carrera, pudiendo finalizar por

primera vez un proceso de desarrollo de software, con toda la documentación necesaria.

Por último, he de decir que este proyecto me ha iniciado en un área que no conocía en profundidad, la inteligencia artificial. Para mí, ha supuesto una oportunidad única para aprender sobre este tema ya que no conocía las opciones existentes de aprendizaje automático para el reconocimiento facial. También he podido experimentar cómo funcionaban algunos de ellos al poner en prueba distintas tecnologías hasta escoger la definitiva.

En resumen, puedo decir que el proyecto ha supuesto una gran oportunidad para mi desarrollo tanto profesional como personal y que gracias a él, he adquirido conocimientos en múltiples áreas que desconocía.

7. Líneas futuras

En este apartado expondré las limitaciones actuales del proyecto y algunas mejoras que se podrían implantar en el futuro.

- En primer lugar, se podría desarrollar más la interfaz para crear una que incorporara el text to speech en todas sus pantallas, ya que de esta forma no dependeremos de la tecnología del móvil del usuario.
- En segundo lugar, se podrían buscar alternativas para la forma en la que se llevarán a cabo los paseos ya que según se encuentra planteada actualmente, la aplicación requeriría que el usuario llevara encima un miniordenador y una cámara. Sin embargo, se han encontrado gafas de sol con cámaras integradas que supondrían una mejora en la facilidad para los usuarios ya que no tendrían que llevar la cámara encima.
- Por último, se podrían incorporar nuevas funcionalidades a la aplicación como una de geolocalización que permitiera guardar la ubicación de los

usuarios mientras llevan a cabo un paseo y guardara los datos de ubicación del usuario al final de este.

8. Referencias

[1] Instituto Nacional de Estadística. (s. f.). INE. Recuperado junio 28, 2023, de <https://www.ine.es/dynt3/inebase/es/index.htm?padre=8494&capsel=8754>

[2] BOE-A-2022-17105 Real Decreto 888/2022, de 18 de octubre, por el que se establece el procedimiento para el reconocimiento, declaración y calificación del grado de discapacidad. (s. f.). Boe.es. Recuperado junio 28, 2023, de https://www.boe.es/diario_boe/txt.php?id=BOE-A-2022-17105

[3] Arsfutura. (s. f.). face-recognition: A framework for creating and using a Face Recognition system. Recuperado junio 28, 2023, de <https://github.com/arsfutura/face-recognition>

[4] Kumar, A., Kaur, A. & Kumar, M. (2019). Face detection techniques: a review. Artif Intell Rev, 52, 927–948. <https://doi.org/10.1007/s10462-018-9650-2>

[5] Amazon.com. (s. f.). ¿Qué es el reconocimiento facial? Recuperado junio 28, 2023, de <https://aws.amazon.com/es/what-is/facial-recognition>

[6] IBM Documentation. (2021, August 17). El modelo de redes neuronales. Recuperado junio 28, 2023, de <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model>

[7] Chrislb. (s. f.). By Chrislb - created by Chrislb, CC BY-SA 3.0. Recuperado junio 28, 2023, de <https://commons.wikimedia.org/w/index.php?curid=224555>

[8] Hardesty, L. (n.d.). Explained: Neural networks. MIT News | Massachusetts Institute of Technology. Recuperado junio 28, 2023, de <https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414>

[9] IBM.com. (s. f.). ¿Qué son las redes neuronales recurrentes? Recuperado junio 28, 2023, de <https://www.ibm.com/es-es/topics/recurrent-neural-networks>

[10] Ccama, D. A. S. (s. f.). Aprendizaje Profundo. Amazonaws.com. Recuperado junio 28, 2023, de https://rstudio-pubs-static.s3.amazonaws.com/896010_68e6c7c0995e4d90b75f02961a5d67fb.html

[11] IBM.com. (s. f.). What are Convolutional Neural Networks? Recuperado junio 28, 2023, de <https://www.ibm.com/topics/convolutional-neural-networks>

[12] dshahid. (2019, February 24). Convolutional neural network. Towards Data Science. Recuperado junio 28, 2023, de <https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529>

[13] Maheshwari, H. (2021, May 11). Basic Text to speech, explained. Towards Data Science. Recuperado junio 28, 2023, de <https://towardsdatascience.com/text-to-speech-explained-from-basic-498119aa38b5>

[14] Vue.js. (s. f.). Introduction. Recuperado junio 28, 2023, de <https://vuejs.org/guide/introduction.html#:~:text=on%20VueMastery.com-%2DWhat%20is%20Vue%3F,be%20they%20simple%20or%20complex.>

[15] Abramowski, N. (2022, July 4). What is NPM? The Complete 2023 Beginner's Guide. CareerFoundry. Recuperado junio 28, 2023, de <https://careerfoundry.com/en/blog/web-development/what-is-npm/>

[16] Seidor.com. (s. f.). ¿Qué es Firebase? ¿Qué ventajas ofrece en 2023 a nuestras apps? Recuperado junio 28, 2023, de <https://www.seidor.com/blog/firebase-que-es>

[17] Amazon.com. (s. f.). ¿Qué es Python? Recuperado junio 28, 2023, de <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes>.

[18] Quiza, J. (2018, June 12). Deep Learning con Pytorch - Ciencia y Datos - Medium. Ciencia y Datos. Recuperado junio 28, 2023, de <https://medium.com/datos-y-ciencia/deep-learning-con-pytorch-672469c637f4>

[19] NVIDIA Data Science Glossary. (s. f.). PyTorch. Recuperado junio 28, 2023, de <https://www.nvidia.com/en-us/glossary/data-science/pytorch/>

[20] Moreno, M. (s.f.) Modelos de proceso de Software. Universidad de Salamanca. Recuperado junio 28, 2023, de <http://avellano.usal.es/~mmoreno/ASTema2.pdf>