

ANEXO I

1. CREACIÓN DE LA APP

El presente proyecto no es la propuesta de una idea, sino que es la muestra de la elaboración de la misma. A continuación se describe parte del largo proceso técnico de creación del diseño y desarrollo de la app Signo, y se muestra las intenciones de futuro con el mismo.

1.1. REQUERIMIENTOS Y LIMITACIONES TÉCNICAS

La elaboración de un proyecto de estas características supone encontrarse con una gran cantidad de problemas y dificultades.

En primer lugar, y puesto que se partía de un conocimiento nulo en materia de desarrollo, ha sido necesario aprender de cero Swift, el lenguaje de programación para iPhone y iPad. Este hecho retrasó el inicio del proyecto dos meses aunque, una vez empezado, el proceso de desarrollo fue más lento de lo habitual, ya que se compaginó el aprendizaje y el desarrollo de manera simultánea para tratar de avanzar lo más rápido posible. Además, Swift es un lenguaje joven y que se actualiza año a año por lo que, pese a haber una cantidad considerable de documentación y recursos en Internet, gran parte se encontraba desactualizada y enfocada versiones antiguas que ya no son compatibles. Y de manera muy similar al lenguaje de programación, se ha tenido que aprender la herramienta Firebase, proporcionada por Google y empleada para el registro de usuarios y la creación de bases de datos. Al igual que en el caso anterior, gran parte de la documentación de Firebase estaba desactualizada.

Entre las necesidades de aprendizaje se encuentra también la de los programas a utilizar, principalmente Sketch para el diseño y Xcode para el desarrollo. Aunque la curva de aprendizaje del primero fue suave, no lo fue tanto la del segundo. Xcode es el entorno de desarrollo proporcionado por Apple para realizar las aplicaciones para sus plataformas, un software complejo por ofrecer gran multitud de opciones y posibilidades.

Finalmente, destacar la inversión económica realizada. Aunque el programa proporcionado por Apple para el desarrollo de aplicaciones es gratuito, no lo es el de diseño, Sketch. Además, publicar aplicaciones en el App Store requiere formar parte del programa de desarrolladores de Apple, con un precio anual de 99€. Por último, ha sido necesaria la compra de un dominio y *hosting* para la landing page, así como un correo personalizado para el soporte de la app y demás necesidades, algo necesario y que denota un mayor nivel de profesionalidad. Normalmente esta inversión inicial se ve compensada en el tiempo cuando la app es lanzada, bien con un pago inicial de la misma a la hora de descargarla, con compras dentro de la app, publicidad... pero no es el caso de Signo. Por ser una aplicación que cumple un fin social, se ha querido tomar una perspectiva sin ánimo de lucro, ofreciendo toda la funcionalidad de manera gratuita.

1.2. PROCESO DE DISEÑO

Tras definir la base del proyecto se comenzó con el proceso de diseño. En este momento se eligió el estilo gráfico del proyecto mediante la paleta de color y tipografía a utilizar entre otros elementos y, seguidamente, se comenzó con la creación de las pantallas.

A la hora de diseñar las pantallas se siguieron los conceptos aprendidos de diseño de Interfaces de Usuario así como de Experiencia de Usuario, a los que había que sumar además las Human Interface Guidelines, una serie de normas dictadas por Apple para el correcto diseño de las aplicaciones de su tienda y que deben seguirse sin excepción si se quiere que la app sea publicada.

Para el diseño de diferentes elementos como barras de navegación o campos de texto se empleó el uso de componentes que ofrece Apple en su página web para desarrolladores con el objetivo facilitar el diseño y utilizar los mismos elementos empleados por ellos para hacer la experiencia del usuario más homogénea entre apps. Para los iconos se empleó la biblioteca SF Symbols, creada también por Apple, y que ofrece una multitud de iconos que ya emplea la compañía en su sistema operativo, lo que facilita el entendimiento por parte del usuario al estar familiarizado con ellos. Todos estos elementos se adaptaron con las líneas de diseño creadas para la app de modo que, aunque los elementos fuesen los creados por Apple, la experiencia fuese personalizada y adaptada al proyecto.

En paralelo al proyecto se creó el logo, que sería utilizado como icono. Aunque se siguieron diferentes propuestas, finalmente se decidió emplear la letra "S", por ser la inicial del nombre, sustituyendo las serifas de la letra por unas comillas, indicando esa voluntad de iniciar una conversación y de contar tus experiencias en la app.

Cuando la app llegó a los últimos estados del desarrollo, se comenzó la elaboración de diseños destinados a la promoción. Se elaboró las imágenes requeridas para la página del App Store, fundamentales para llamar la atención de los visitantes y que descarguen la app, así como las imágenes destinadas a las redes sociales para dar a conocerla.

1.3. PROCESO DE DESARROLLO

Una vez se conoce el programa y se ha aprendido el lenguaje de programación, puede comenzar el desarrollo.

La app Signo hace uso de diferentes *frameworks* o librerías nativas de Swift, siendo la más utilizada en el proyecto MapKit. Pese a que también estaba la opción de emplear los mapas de Google, utilizados en una gran variedad de aplicaciones, el hecho de que la app se desarrolle exclusivamente para iPhone ha hecho que se considere como mejor la opción de emplear los mapas de Apple, cada vez también más utilizados. MapKit es un *framework* muy completo que proporciona todo lo relativo al uso de mapas y que, por lo tanto, encajaba a la perfección con la aplicación.

Una vez implementada la funcionalidad del mapa se procedió a realizar el desarrollo de los botones de la pantalla principal y su correspondiente funcionamiento. Entre esos botones encontramos el de búsqueda, el de ubicación del usuario y el de cambiar el tipo de mapa, además del botón del menú lateral y del botón de filtrado de agresiones.

Además de con librerías nativas se ha trabajado con Cocoapods, librerías externas creadas por otros usuarios que pueden implementarse en el proyecto. Esto tiene la finalidad de facilitar el proceso de desarrollo al implementar elementos cuyo código es extenso o complicado de elaborar, lo que facilita enormemente la realización del proyecto. Entre los *pod*s utilizados en Signo se encuentra el menú lateral, el botón de filtrar o el panel que aparece cuando mantienes pulsado en el mapa para añadir una agresión. Sin embargo, es necesario indicar que se tiene la intención es desarrollar de manera nativa estos elementos en un futuro para no tener que depender del código y del mantenimiento del mismo por parte de otros usuarios.

Una de las partes más importantes de la app es la función de añadir un signo. Para ello se ha trabajado con el reconocimiento de gestos, ya que el panel se activa al mantener pulsada la pantalla durante unos segundos. Además se ha hecho uso de protocolos, una forma de comunicación entre pantallas que era necesaria para que en la nueva pantalla de añadir una agresión apareciese el nombre de la calle que se había seleccionado previamente en el mapa. El uso de protocolos es complejo y se ha empleado en diversas pantallas, por lo que aprender su funcionamiento ha necesitado un mayor tiempo de estudio e implementación.

Otro de las funciones imprescindibles de la app es el registro e inicio de sesión de los usuarios. Puesto que para añadir una agresión los usuarios deben estar identificados, era necesario añadir la funcionalidad de creación de cuentas, que se ha llevado a cabo mediante Firebase. Firebase es una herramienta hecha por Google y facilita las tareas de registro de usuarios y creación de base de datos, entre otras muchas funciones. Esto ha constituido otro punto importante en el desarrollo, puesto que ha sido necesario aprender el funcionamiento de esta herramienta y, al igual que con el lenguaje Swift, la mayor parte de la documentación que puede encontrarse está desactualizada debido al gran ritmo de actualizaciones.

Con Firebase se ha elaborado también las bases de datos: una para el listado de usuarios registrados y cinco para los diferentes tipos de agresiones. La más compleja es la de los usuarios puesto que, además de tener la información de registro, tiene dos bases de datos internas: la de los signos que ha añadido ese usuario y los borradores que ha guardado para publicar en otro momento. Se ha empleado Firebase Firestore,

base de datos en tiempo real y con una estructura de colección-documento-colección, en contraposición a Firebase Realtime, la primera base de datos ofrecida por Firebase y que no se ajustaba tanto a las necesidades del proyecto.

Una vez la aplicación estaba en una fase de desarrollo casi final, se procedió a anunciar la primera beta, que se publicó con la herramienta Testflight y que fue instalada por un grupo de usuarios cercano y de confianza con el objetivo de que reportaran los fallos que se fuesen encontrando. Esto sirvió para adquirir un primer *feedback* que mejoró algunas de las funcionalidades de la app a la vez que se trabajaba con otras nuevas que debían estar antes de que la app fuese lanzada oficialmente.

Finalmente, cuando la app estuvo completa y sin fallos aparentes, se mandó a revisión al equipo de Apple. Las aplicaciones pasan un examen muy estricto por parte de Apple, ya que tienen que ceñirse a las Human Interface Guidelines, normas que pone la compañía para que todas las aplicaciones publicadas en su tienda cumplan unos estándares de calidad. Tras ser rechazada las dos primeras ocasiones, se corrigieron los fallos vistos por Apple y fue finalmente admitida para publicación el 12 de julio de 2021.

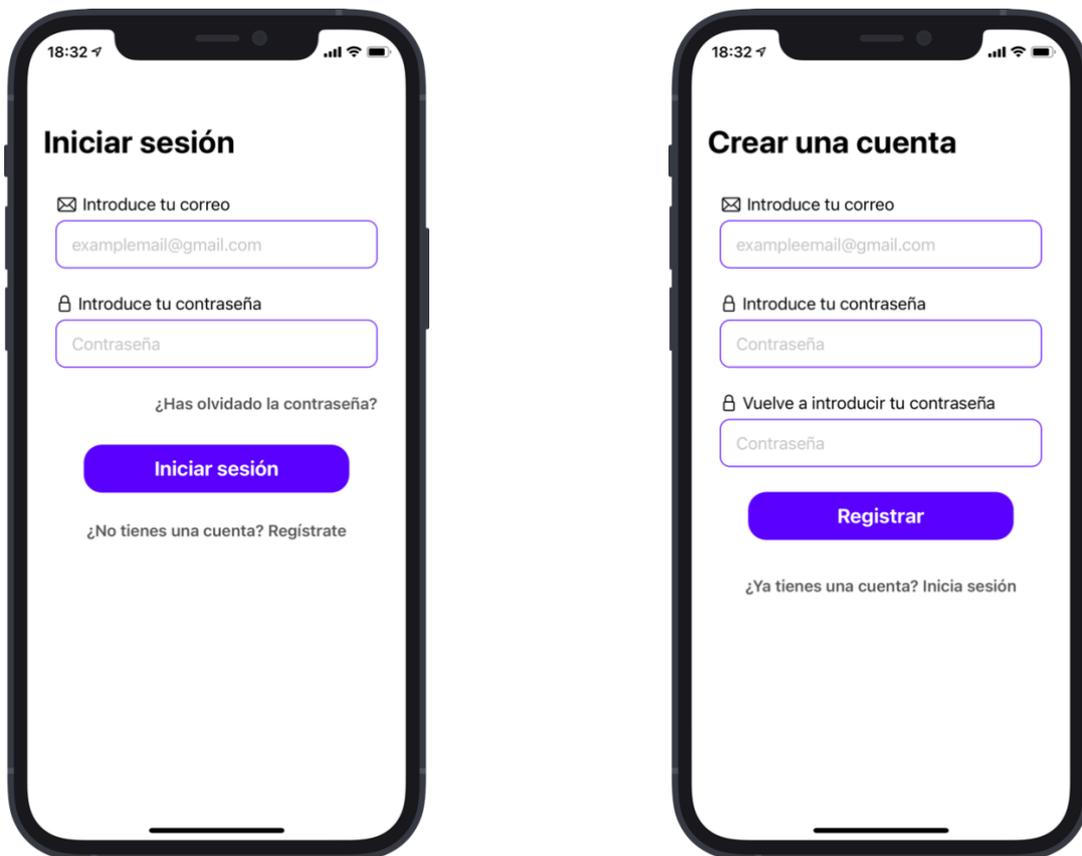
1.4. PRÓXIMOS PASOS EN EL DESARROLLO

La complejidad de la app no reside únicamente en su fase de desarrollo, sino en el hecho de tener que mantener el código con el paso de los años, adaptándolo a las nuevas versiones del sistema operativo, a la vez que actualizar la app con nuevas características y funcionalidades, lo que hace que este proyecto no tenga fin.

Tal y como se explica en el Trabajo Fin de Grado, para el proyecto inicial se trabajó con el concepto de Producto Mínimo Viable (MVP; Minimum Viable Product). Esta aproximación supone crear una primera versión del proyecto con el número mínimo de características, únicamente las imprescindibles para su correcto funcionamiento y, de esta forma, poder lanzarlo al mercado con la mayor brevedad. Una vez el proyecto esté disponible para todos, se trabajará de manera constante para ofrecer nuevas funciones y eliminación de errores para mantenerlo siempre en las mejores condiciones.

2. PANTALLAS DE LA APP

Aunque lo recomendable es descargarla para seguir el desarrollo del trabajo de la manera más completa, se muestra a continuación para aquellas personas que no hayan podido hacerlo.



Pantalla de inicio de sesión (izquierda) y de registro (derecha).



Pantalla principal (izquierda) y botón de filtrado extendido (derecha).



Menú de añadir una agresión recogido (izquierda) y mismo menú extendido (derecha).



Ejemplo de agresión introducida (izquierda) y pantalla de edición de un Signo (derecha).