



# APP VOLUNTIER

Manual del programador

Autora: Laura Lucena Buendia 2º DAM

|                                          |           |
|------------------------------------------|-----------|
| <b>CAPÍTULO 1: INTRODUCCIÓN</b>          | <b>3</b>  |
| 1.1. Objetivos y alcance del proyecto    | 3         |
| 1.2. Desglose temporal del proyecto      | 4         |
| <b>CAPÍTULO 2: ANÁLISIS Y REQUISITOS</b> | <b>5</b>  |
| 2.1. Splash                              | 5         |
| 2.2. Login/Registro de usuario           | 5         |
| 2.3. Creación del perfil                 | 6         |
| 2.4. Gestión de tareas                   | 7         |
| 2.5. Puntuaciones                        | 7         |
| <b>CAPÍTULO 3: DISEÑO</b>                | <b>8</b>  |
| <b>CAPÍTULO 4: IMPLEMENTACIÓN</b>        | <b>8</b>  |
| 4.1. Firebase                            | 9         |
| 4.1.1. Firebase Authentication           | 9         |
| 4.1.2. Firebase Firestore:               | 9         |
| 4.2. Mapbox                              | 10        |
| 4.3. GIT                                 | 10        |
| <b>CAPÍTULO 5: CONCLUSIONES</b>          | <b>11</b> |

# CAPÍTULO 1: INTRODUCCIÓN

En este primer capítulo se procederá a una breve introducción del proyecto, objetivos del mismo y la motivación tanto tecnológica como personal para poder ser un proyecto cuyo resultado ayude a las personas en su día a día.

## 1.1. Objetivos y alcance del proyecto

App Voluntier es un proyecto nacido a raíz de la pandemia y la necesidad de ayudar a aquellas personas más vulnerables.

El objetivo es que todas aquellas personas necesitadas de alguna ayuda (como puede ser ir al supermercado pero no poder ir por ser una persona de riesgo o necesitar un medicamento y no está en condiciones de ir a la farmacia) pueda crear esa necesidad y personas voluntarias pueda ayudarla. Establecer una vía de contacto a través de una app y poder conseguir que las personas se ayuden mutuamente es el gran objetivo de este proyecto.

Para ello, este proyecto será desarrollado con Ionic, por los beneficios que aporta a la hora de tener la aplicación en diferentes plataformas (Android/iOS). Se ha utilizado Visual Studio como entorno de desarrollo.

## 1.2. Desglose temporal del proyecto

En esta sección se detalla, a través de la siguiente tabla, el desglose de horas dedicado al proyecto. En cada funcionalidad se ha dedicado tiempo a investigación y problemas derivados del desarrollo.

| Desarrollo (semana)            | 8 Marzo | 15 Marzo | 22 Marzo | 29 Marzo | 5 Abril | 12 Abril | 19 Abril | 26 Abril | 2 Mayo | 10 Mayo | 17 Mayo | 24 Mayo | 31 Mayo | 7 Junio | 14 Junio |
|--------------------------------|---------|----------|----------|----------|---------|----------|----------|----------|--------|---------|---------|---------|---------|---------|----------|
| Definición del proyecto        | 6h      | 2h       |          |          |         |          |          |          |        |         |         |         |         |         |          |
| Diseño gráfico                 |         | 12h      | 12h      | 24h      |         |          |          |          |        |         |         |         |         |         |          |
| Splash                         |         |          | 3h       |          |         |          |          |          |        |         |         |         |         |         |          |
| Proceso de login/registro      |         |          |          | 25h      | 15h     |          |          |          |        |         |         |         |         |         |          |
| Perfil Voluntario/ Solicitante |         |          |          |          |         | 12h      | 10h      | 5h       |        |         |         |         |         |         |          |
| Creación y gestión de tareas   |         |          |          |          |         |          |          | 4h       | 8h     | 8h      |         |         |         |         |          |
| Mapa                           |         |          |          |          |         |          |          |          |        |         | 10h     |         |         |         |          |
| Puntuaciones                   |         |          |          |          |         |          |          |          |        |         |         | 6h      | 4h      |         |          |
| Documentación                  |         |          |          |          |         |          |          |          |        |         |         |         |         | 10h     | 6h       |
| Presentación                   |         |          |          |          |         |          |          |          |        |         |         |         |         |         | 8h       |

TOTAL => 190 Horas

## CAPÍTULO 2: ANÁLISIS Y REQUISITOS

En este capítulo se detalla los requisitos que tendrá la aplicación. Dichos requisitos están agrupados por grandes funcionalidades, donde cada una de ella se componen de varios casos de usos posibles.

En relación a todos los casos de uso, los actores que pueden intervenir son:

- Usuario Voluntario: usuario dado de alta en la aplicación, cuyo perfil consiste en escoger, realizar y puntuar tareas de usuarios Solicitantes que exponen la necesidad que tienen.
- Usuario Solicitante: usuario dado de alta en la aplicación, cuyo perfil consiste en crear tarea y puntuar a usuarios Voluntarios.

### 2.1. Splash

Caso de uso que representa el inicio de la aplicación. Dicha pantalla es transitoria y se navegará a una sección u otra dependiendo de las siguientes casuísticas:

- Si el usuario no está autenticado, se navegará a la sección de login/registro de usuario.
- Si el usuario está autenticado pero no tiene un perfil creado (Voluntario o Solicitante), se navegará al proceso de creación del perfil.
- Si el usuario está autenticado y tiene un perfil creado (Voluntario o Solicitante), se navegará a la pantalla principal dependiendo de su perfil.

## 2.2. Login/Registro de usuario

Funcionalidad que representa el acceso a la aplicación, ya sea autenticándose porque el usuario ya tiene una cuenta creada previamente o bien registrándose por primera vez. Independientemente de esto, existen varios casos de usos para completar esta funcionalidad:

- Caso de uso login/registro con Google: el usuario podrá autenticarse o registrarse en la aplicación a través de Google siempre y cuando tengas una cuenta de dicho proveedor.
- Caso de uso login/registro con Facebook: el usuario podrá autenticarse o registrarse en la aplicación a través de Facebook siempre y cuando tengas una cuenta de dicho proveedor.
- Caso de uso login/registro con Apple: el usuario podrá autenticarse o registrarse en la aplicación a través de Apple con Apple Id siempre y cuando tengas una cuenta de dicho proveedor.
- Caso de uso login/registro con email /contraseña: el usuario podrá autenticarse o registrarse en la aplicación a través de un email y una contraseña.
- Caso de uso cerrar sesión: el usuario podrá cerrar sesión independientemente del caso de uso de login realizado.

## 2.3. Creación del perfil

Una vez que el usuario se haya autenticado/registrado en la app y no tenga un perfil creado, se procederá con dicha funcionalidad.

Los casos de usos que se ejecutarán serán los siguientes:

- Elección del tipo de usuario: el usuario elegirá entre ser Voluntario o Solicitante.
- Rellenar formulario de datos personales.
- Rellenar formulario de habilidades/necesidades según su tipo de perfil.

Una vez completado estos pasos, el perfil se creará asignado a ese usuario. No tendrá que volver a rellenar este perfil, a menos que desee modificar algún campo de estos en la edición de su perfil.

## 2.4. Gestión de tareas

Esta funcionalidad representa todo lo relacionado con tareas: tareas completadas, pendientes etc.

Un usuario Voluntario tiene un buscador para buscar Tareas pendientes de realizar, pudiendo así asignársela y poder realizarla. Una vez completada, podrá recibir una puntuación del Solicitante.

Un usuario Solicitante publicará Tareas que necesita realizar a través de un Voluntario.

Los casos de uso relacionados con esta funcionalidad son:

- Creación de una tarea: un Solicitante crea una Tarea con toda la información necesaria para poder realizarla.
- Búsqueda de tareas: un Voluntario podrá buscar, a través de un buscador, tareas cercanas que pueda realizar. Solo estarán visibles aquellas que están pendientes de realizar y no hay ningún Voluntario asignado a la tarea.
- Completar tarea: un Solicitante dá la tarea por completada cuando el Voluntario la ha completado, para verificar su correcta finalización.

## 2.5. Puntuaciones

Un Voluntario podrá puntuar a un Solicitante y viceversa, a través de una Tarea completada. Con ello, se quiere conseguir reflejar en la aplicación la valoración de todos los usuarios y tenerlo en cuenta.



## CAPÍTULO 3: DISEÑO

El diseño gráfico de la aplicación ha sido elaborado manualmente y totalmente personalizado para este proyecto. Para ello se han utilizado las herramientas software Adobe Photoshop y ProCreate. Los elementos hardware utilizados han sido un iPad y un tablet Wacom de diseño gráfico.

## CAPÍTULO 4: IMPLEMENTACIÓN

En este capítulo entraremos en el detalle de la implementación del proyecto.

Cómo bien se dijo en la introducción, el proyecto ha sido desarrollado con Ionic, concretamente con Ionic 5. Como entorno de desarrollo se ha utilizado Visual Studio Code, ya que tiene todo lo necesario para desarrollar el proyecto sin depender de otros entornos.

Se ha utilizado el Framework *Capacitor* para “transformar” la app en aplicaciones Android/iOS mediante el uso de Webview. Se ha decidido utilizar este Framework por varios motivos:

- Nos ofrece la exportación en un proyecto en Android Studio (Android) y XCode (iOS).
- Facilidad de integración y compatibilidad con todos los plugins de Cordova e Ionic
- Se integra completamente con Angular. En el momento de compilar, no es necesario hacer ajustes en los scripts como es el caso de Cordova.

Una vez determinado qué lenguaje/entorno y gestión de plugins se ha utilizado, se procederá a explicar las librerías utilizadas en el proyecto.

## 4.1. Firebase

Una gran opción para varias partes de la aplicación ha sido la elección de Firebase, dado el soporte que tiene, oferta de librerías amplia y su gratuidad en algunos de sus servicios.

Las librerías de Firebase que se ha utilizado son:

### 4.1.1. Firebase Authentication

Se ha utilizado esta librería para la gestión del registro/login en la aplicación. Gracias a esta librería, podemos gestionar los 4 proveedores de login que se ofrecen en la app, así como el proceso de cierre de sesión.

### 4.1.2. Firebase Firestore:

Se ha utilizado esta librería para base de datos compartida entre las aplicaciones/usuario. En ella se almacenan todos los usuarios, perfiles de los mismos, tareas y puntuaciones.

Esta base de datos es una base de datos no relacional/noSQL. En ella no existen tablas ni filas. En su lugar, tenemos las denominadas “Colecciones” que pueden identificarse como las tablas de una base de datos SQL. Estas “Colecciones” tienen “Documentos”, que contienen una serie de datos almacenados. Cada “Documento” se puede identificar con una fila de una tabla en base de datos SQL. Cada documento tiene información guardada sobre un modelo de datos creado en la app.

La flexibilidad y escalabilidad de esto es que en el proyecto se crean las interfaces necesarias y Firebase Firestores almacena todos los atributos tal cual dentro de un documento. No hay que definir la estructura en Firebase

como sí pasa con las bases de datos SQL. Es el propio front el que decide que se guarda en cada Colección y los tipos de datos.

## 4.2. Mapbox

Para la implementación del mapa de la aplicación se estudiaron 3 posibles librerías:

- **Google Maps:** la librería por excelencia de mapas, con el soporte de Google y todos sus beneficios. El problema de esta parte es que tiene facturación (aunque con X peticiones gratuitas cada mes) y obligado a poner una dirección y tarjeta de crédito para dicha facturación. Por lo tanto, dado la imposibilidad de esto último, se descartó esta librería.
- **Here Map** (<https://developer.here.com/>): esta librería, menos conocida, es otra librería de mapas con bastante soporte, ya que muchos de los GPS integrado de los coches (marcas como Audi lo implementan). Se decidió utilizarla pero dada la poca documentación y foros de esta librería, no se pudo continuar con ella, descartando su uso final.
- **Mapbox:** finalmente, Mapbox fue la librería elegida. Mapbox es una librería utilizada en Angular/Ionic y con bastante documentación/foros para su implementación. A este se le une su gratuidad, sin coste alguno para este proyecto y su fácil integración en el proyecto. Además, su consola de gestión es bastante intuitiva.

### 4.3. GIT

Para la gestión del proyecto se ha utilizado el Sistema de Control de Versiones (SCV) GIT junto a la plataforma Bitbucket, perteneciente a Atlassian.

## CAPÍTULO 5: CONCLUSIONES

Se puede concluir que el proyecto ha sido desarrollado con éxito aunque se han encontrado algunas limitaciones/observaciones durante el proyecto:

- Implementación de login con Apple, login con Facebook y Google Maps: no ha sido posible estas implementaciones debido a su obligatoriedad de vincular una tarjeta de crédito para posibles facturaciones. En el caso de Apple un paso más allá, pagar las tasas de desarrollador anual.
- Login con Apple: aparte de lo anterior, se ha decidido al menos incluirlo, dado que Apple obliga a implementar su propio login a todas las apps que ofrecen login de terceros (Google, Facebook, Twitter, etc). Más información aquí <https://developer.apple.com/app-store/review/guidelines/#sign-in-with-apple>
- Al tener un **diseño tan personalizado y hecho a mano**, ha consumido un gran tiempo/esfuerzo, pero no hay mejor manera de llegar a un usuario que haciendo una atractiva aplicación.