Un dibujo con letras

Descripción generada automáticamente con confianza media

**Icono

Descripción generada automáticamente**

## SUNRISE WEATHER APP

**Tutora:** Raquel Cerdá

**Ciclo Formativo:** DAM

### Carlos Manuel Alcántara Rodríguez

Alfredo de los Ríos Carrizo

Alberto Lozano Gómez

1. INTRODUCCIÓN

La finalidad de este primer capítulo es proporcionar al lector una visión general

de este proyecto, explicando su contexto, motivación, los objetivos planteados y la forma en que se espera que sean satisfechos.

* 1. CONTEXTO

El presente trabajo forma parte de la etapa final del Ciclo de Grado Superior en Desarrollo de Aplicaciones Multiplataforma. Para ello, este grupo ha realizado una aplicación móvil que informe al usuario sobre la meteorología de un determinado lugar, accediendo a los datos proporcionados por la empresa AccuWeather.

* 1. MOTIVACIÓN

Después de varias reuniones de grupo, aportando ideas y posibles tecnologías para llevarlas a cabo, nos dimos cuenta de que nos apetecía mucho implementar lo que hasta ahora habíamos aprendido, pero también experimentar. Por ello, nos decantamos por realizar un proyecto de una aplicación para móvil y hacer el desarrollo con el lenguaje Kotlin y utilizando el kit de herramientas moderno de Android, Jetpack Compose.

El objetivo principal de este proyecto no es solo aprender más sobre usabilidad, experiencia de usuario, diseño de interfaces, etc. sino también conocer una reciente tecnología, experimentar con ella y aprender una nueva disciplina.

Siendo conscientes de nuestras limitaciones, aportamos proyectos que pudiéramos abarcar. La idea de proyecto surgió fruto de una votación grupal, siendo una app meteorológica la finalmente elegida. Nos pareció correcto realizar algo tan cotidiano, accesible y recurrente como es una aplicación que nos pudiera reportar datos meteorológicos al instante de una determinada zona.

El objetivo final es aprender y sentirnos satisfechos del trabajo realizado.

1. PALABRAS CLAVE

**Android**: Sistema operativo móvil basado en el núcleo Linux y otros softwares de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil.

**Kotlin:**  Es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de Java. Aunque no tiene una sintaxis compatible con Java, está diseñado para interoperar con código Java y es dependiente de su biblioteca de clases, tal como pueda ser el entorno de colecciones de Java. Conocido por ser utilizado en la creación de aplicaciones de Android.

**Jetpack Compose**: Es un nuevo kit de herramientas moderno para la creación de vistas de una forma declarativa en Android, el cual simplifica y acelera el desarrollo de la IU haciendo que las apps cobren vida rápidamente con menos código, gracias a la reutilización de componentes.

#### Composable: Jetpack Compose se basa en funciones componibles. Estas funciones le permiten definir la interfaz de usuario de su aplicación mediante programación describiendo cómo debería verse y proporcionando dependencias de datos, en lugar de centrarse en el proceso de construcción de la interfaz de usuario (inicializar un elemento, adjuntarlo a un padre, etc.). Los composables pueden ser reutilizados agilizando así el tiempo de desarrollo.

**API:** Application Programming Interfaces. Se trata de un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la comunicación entre dos aplicaciones de software a través de un conjunto de reglas. Establece cómo un módulo de un software se comunica o interactúa con otro para cumplir una o muchas funciones. Siempre dependiendo de los permisos que dé el propietario de la API a los desarrolladores de terceros.

**Retrofit:**  Es un cliente de servidores REST para Android y Java, compatible con Kotlin y desarrollado por Square. Permite hacer peticiones al servidor y gestionar diferentes tipos de parámetros, parseando automáticamente la respuesta a un tipo de dato.

1. MÓDULOS FORMATIVOS APLICADOS

* Programación
* Entornos de desarrollo
* Desarrollo de Interfaces
* Programación multimedia y dispositivos móviles
* Acceso a datos
* Programación de servicios y procesos
* Inglés

1. HERRAMIENTAS / LENGUAJES UTILIZADOS

* **Android Studio** es el IDE oficial de Android.
* **Kotlin** como lenguaje de programación.
* **Jetpack Compose** para el diseño de la UI.
* **Figma** es una herramienta de diseño que se aloja en la web, la cual utilizamos para el prototipo de la UI y de la navegación.
* **API AccuWeather** nos proporciona los datos meteorológicos necesarios.
* **Retrofit2** como herramienta para conectar con la API mediante un servicio REST.
* **JSON** es el formato de respuesta de datos de la API.
* **Quicktype**
* **Moshi,** es una librería para parsear JSON en objetos Java o Kotlin.
* **Postman** nos permite realizar peticiones de una manera simple para testear APIs de tipo REST.
* **OkHttp** es un cliente HTTP. Cuando hay problemas de conexión, mediante hilos intentará direcciones alternativas para preservar la red.
* **Coroutine:** Permiten escribir código asíncrono simple y mantiene la capacidad de respuesta de la app mientras administra tareas prolongadas como operaciones complejas o muchas llamadas simultáneas a la API.
* **Coil** es una biblioteca de carga de imágenes para Android y respaldada por Kotlin Coroutines. Utilizada para la descarga de los iconos de la API.
* **Git** para el control de las diferentes ramas y versiones.
* **Discord** como servicio de mensajería instantánea y reuniones de grupo.

1. COMPONENTES DEL GRUPO

* **Alfredo de los Ríos Carrizo**

La aportación realizada en el proyecto ha sido la creación a los tres rest api utilizados en el proyecto: el rest api a **AccuWeather, para recolectar los datos** meteorológicos **y los rest API de** <https://api.myip.com> y [*http://ip-api.com/*](http://ip-api.com/)**para sustituir la localización por gps. Tambien ha realizado la adaptación de datos del rest API de AccuWeather al modelo de datos utilizado por la aplicación.**

* **Alberto Lozano Gómez**

La aportación realizada en el proyecto principalmente está enfocada al diseño e inserción de datos en la UI. Desarrollando los distintos *composables* de la interfaz, ajustándose al diseño inicial.

* **Carlos Manuel Alcántara Rodríguez**

1. FASES DEL PROYECTO



###### Tormenta de ideas

Diseño en Firgma

En las reuniones previas de equipo, desarrollamos un plan de estudio de los requerimientos y necesidades para llevar a cabo el proyecto.

Tomamos como referencia la app que por defecto viene instalada en nuestros teléfonos móviles y a partir de ahí diseñamos nuestra interfaz con *Figma* y analizamos los datos que debemos solicitar a la API para llevar a cabo nuestra idea de proyecto.

###### Texto Descripción generada automáticamenteRecopilación de datos

Elegimos [AccuWeather API](https://developer.accuweather.com/) como fuente de datos meteorológicos.

Para acceder a los diferentes rest API que utilizamos en la aplicación utilizamos retrofit, utilizando Moshi para pasar los json a clases, tomando como referencia <https://cursokotlin.com/tutorial-retrofit-2-en-kotlin-con-corrutinas-consumiendo-api-capitulo-20-v2/> . En la interface QuotesApi\_Tiempo se ha puesto las llamadas a los puntos de acceso que en un momento dado se podría acceder para el desarrollo de la aplicación.Se ha construido el objecto RetrofitHelper para implementar la interface. Esté se utiliza en un objecto Acceso\_API, el cual a través de uno de las entradas para la busqueda de datos, coger\_tiempo\_posicion\_gps, coger\_tiempo\_nombre y coger\_tiempo\_key, recoge los datos de la petición de la vista. Estos tres métodos llaman a la función privada coger datos.

Está función selecciona la ciudad dependiendo de los datos introduccidos y recolecta los datos de la rest API de AccuWeather, todo esto ejecutado en una corrutina.

A continuación, en la misma fución, se pasa los datos recolectados a un atributo de la clase perteneciente a la clase Datos\_Tiempo, cuya utilidad es pasar los datos a la vista. Y por último se activa la bandera de datos\_adquiridos, para que la vista sepá que ya tiene los datos disponibles.

###### Icono Descripción generada automáticamenteDiseño de la UI

Aplicamos Jetpack Compose para el diseño de vistas.

La interfaz de usuario consta de varios composables:

* Splash
* SearchBar
* Daily Card
* Details
* Hourly 12H
* Next 5 days



Splash

Es una screen con un diseño sencillo. Contiene el logo de la aplicación y un breve texto.

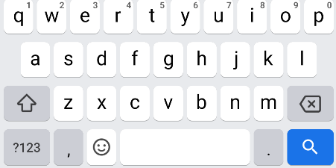
Tiene un delay de 4 segundos, pasados estos, navegamos directamente a la pantalla principal.

SearchBar

Es un appBar con función de buscador.

En un inicio nos presenta el nombre de la aplicación y un icono con forma de lupa, que al presionar accedemos al buscador.

La searchBar consta de dos estados: Open y Closed, accediendo a cada uno de ellos con la o con . El estado por defecto es Closed.

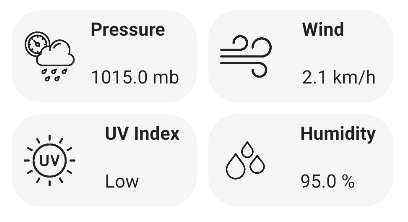
El modo Open dispone de tres funciones que controlan su interactividad: la mutación del texto de entrada, cuando se da a la lupa del teclado que captura lo escrito en el campo de texto o cuando se presiona para cerrar el buscador.

Daily Card

Es la tarjeta de información principal de la interfaz. En ella se muestran los principales datos meteorológicos del día:

* Nombre de la localidad
* Fecha del día
* Temperatura actual
* Temperatura mínima y máxima
* Icono diario
* Frase de previsión meteorológica del día

Details

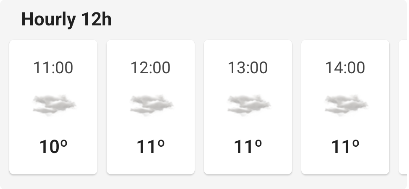
Se muestra en detalle datos más técnicos referentes a las condiciones meteorológicas diarias.

* Presión atmosférica
* Velocidad del viento
* Índice de UV
* Humedad

Hourly 12H

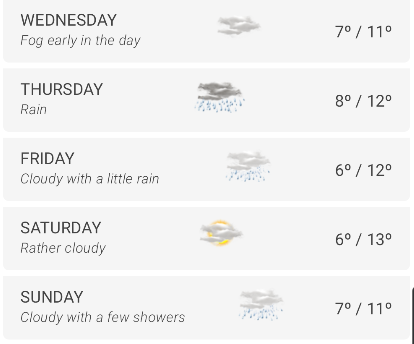
Mediante este composable informamos de las condiciones meteorológicas del día fragmentadas en las próximas 12 horas. La presentación de los diferentes componentes se hace mediante un LazyRow, consiguiendo un scrooll horizontal.

Los datos que se presenta cada componente son:

* Hora
* Icono de cada hora
* Temperatura de cada hora

Next 5 days

Este parte de la interfaz dispone de cinco composables, cada uno referente a uno de los cinco días posteriores a la fecha de la consulta.

Los datos ofrecidos son los siguientes:

* Día de la semana escrito en letra
* Frase de previsión meteorológica
* Icono diario
* Temperatura mínima
* Temperatura máxima

Pantalla Principal Completa

