# 2. Requisitos del prototipo a implementar

El objetivo del proyecto es comparar la implementación de un mismo prototipo de sistema utilizando dos tecnologías diferentes (Android y iOS).

Es importante cumplimentar este apartado antes de empezar a implementar el prototipo de cada tecnología, porque ambos prototipos deben cumplir los requisitos que se establezcan en este apartado. El contenido de este apartado es lo que han de compartir ambos equipos como punto de partida.

Se puede dar libertad a los equipos de desarrollo en cuanto al diseño, pero la funcionalidad debe ser lo más parecida posible.

## 2.1 Requisitos funcionales

En la siguiente tabla se indicará el catálogo de requisitos funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RF001 | La definición de datos se basará en un archivo en formato XML. |
| RF002 | La aplicación ha de recibir los datos directamente de internet, sin ningún tipo de almacenamiento. |
| RF003 | La aplicación mostrará por pantalla la información del tiempo incluyendo la temperatura y las precipitaciones. |
| RF004 | La aplicación ha de ser capaz de mostrar un icono en relación a la descripción del tiempo obtenida. |
| RF005 | El sistema ha de poder mostrar el tiempo de una determinada ubicación geográfica. |
| RF006 | La temperatura se mostrará en ºC. |
| RF007 | Al pulsar un selector, se alternará en día/noche (color). |
| RF008 | La aplicación realizará una búsqueda de datos única y seleccionará la hora adecuada en correspondencia con el sistema. |
| RF009 | Existirá un botón que permita la actualización de los datos. |

## 2.2 Otros requisitos

En este apartado se incluyen todos los requisitos no funcionales, de tipo seguridad, interfaz, y otros tipos.

En la siguiente tabla se indicará el catálogo de requisitos no funcionales del sistema.

| **REQ.** | **DESCRIPCIÓN** |
| --- | --- |
| RI001 | El sistema ha de soportar una layout mínima lista. |
| RI002 | La aplicación ha de tener una interfaz de usuario utilizable por personas con daltonismo. |
| RI003 | La interfaz debe utilizar un tamaño y un tipo de fuente totalmente legibles por los usuarios de la aplicación. |
| RI004 | Los colores de la interfaz no han de ser demasiado intensos ni demasiado claros, con el fin de que sea lo más agradable posible. |
| RL001 | El sistema ha de desarrollarse en Java (Android) y en Swift (iOS). |
| RS001 | El sistema almacenará adecuadamente los datos de los usuarios con la seguridad necesaria. |
| RR001 | Todas las respuestas del sistema han de producirse en un máximo de 30 segundos. |
| RR002 | Adquirir la información meteorológica no supone ningún coste. |
| RR003 | La aplicación no ha de tardar más de 5 segundos en iniciarse. |
| RU001 | Los usuarios han de aprender a utilizar la aplicación en un minuto. |
| RU002 | La aplicación debe tener una navegación fácil y suave a través de sus pestañas. |

# 3. Criterios de comparación en la implementación

En el trabajo TG2 se definieron criterios de comparación de las dos tecnologías a nivel teórico.

En este trabajo hay que definir criterios para la comparación de la implementación de las tecnologías en la construcción del prototipo de sistema de ejemplo, cuyos requisitos son los establecidos en el apartado 2.

Se trata de criterios del tipo” “horas empleadas en el desarrollo del sistema”, “velocidad de funcionamiento del sistema”, “recursos necesarios”, etc.

## 3.1 Criterio 1: Tiempo de creación de la interfaz de usuario

Se trata de ver el total de horas que se han necesitado para conseguir una interfaz de usuario funcional en el sistema iOS y en el sistema Android.

Tipo de valor (iOS): numérico (horas)

Tipo de valor (Android): numérico (horas)

## 3.2 Criterio 2: Tiempo de aprendizaje

Horas invertidas en el aprendizaje de los desarrolladores en iOS. Se incluye en este criterio tanto el tiempo de aprendizaje en el entorno de desarrollo como el tiempo invertido en estudiar los lenguajes de programación necesarios.

Tipo de valor (iOS): numérico (horas)

Tipo de valor (Android): numérico (horas)

## 3.3 Criterio 3: Tiempo de instalación del emulador

Este criterio se corresponde con el total de tiempo invertido en la instalación de los emuladores del sistema correspondiente para evaluar qué sistema ha sido más rápido en este aspecto.

Tipo de valor (iOS): numérico (horas)

Tipo de valor (Android): numérico (horas)

## 3.4 Criterio 4: Soporte del entorno de desarrollo

Se trata de ver si los entornos de desarrollo que se han utilizado para el desarrollo de los sistemas poseen un soporte en caso de que el programador necesite consultar alguna duda, o para resolver incidencias

Tipo de valor (iOS): booleano (Si/No)

Tipo de valor (Android): booleano (Si/No)

## 3.5 Criterio 5: Coste del entorno de desarrollo

Evaluar si se ha necesitado desembolsar en algún momento un capital a cambio de poder acceder al entorno de desarrollo (pago de licencia), o si por el contrario, el acceso ha sido gratuito desde el primer momento (libre).

Tipo de valor (iOS): económico (euros)

Tipo de valor (Android): económico (euros)

## 3.6 Criterio 6: Coste del emulador

Al igual que el Criterio 5, evaluar el coste económico que han conllevado las pruebas en los emuladores utilizados en el proceso.

Tipo de valor (iOS): económico (euros)

Tipo de valor (Android): económico (euros)

## 3.7 Criterio 7: Calidad del sistema

## Se trata de evaluar la calidad del sistema creado por los desarrolladores en función de sistemas existentes en el mercado, es decir, realizar una comparación de todos los aspectos de cada aplicación para ver en cuál es mejor cada una.

Tipo de valor (iOS): intervalo [0,10]

Tipo de valor (Android): intervalo [0, 10]

## 3.8 Criterio 8: Eficiencia del entorno de desarrollo

Se ha de evaluar la eficiencia de los emuladores para ver aspectos como cuál ha sido el más rápido en el arranque o el más rápido en la compilación. Se valorarán también aspectos como los lenguajes disponibles y comodidades para el desarrollador

Tipo de valor (iOS): intervalo [0,10]

Tipo de valor (Android): intervalo [0, 10]

## 3.9 Criterio 9: Eficiencia del emulador

Al igual que el Criterio 8, se ha de comprobar la eficiencia en el emulador, valorando la rapidez de arranque, la versión del sistema virtual o la posibilidad de pausar la emulación en un estado en concreto

Tipo de valor (iOS): intervalo [0,10]

Tipo de valor (Android): intervalo [0, 10]

## 3.10 Criterio 10: Lenguajes de programación

Este criterio consiste en analizar los lenguajes de programación que se han necesitado para el desarrollo de las aplicaciones en cada sistema, evaluando tanto el número como la dificultad para realizar la implementación correcta en cada uno.

Tipo de valor (iOS): intervalo [0,10]

Tipo de valor (Android): intervalo [0, 10]