



---

## **Nuevas Profesiones**

### **DESARROLLO DE APLICACIONES MULTIMEDIA**

#### **Campus Cámara**

Proyecto Integrado

**Mercurio**

**Miguel Chiara**

Sevilla, Junio de 2021

# Agradecimientos

- Mis padres, por aportarme ideas y soluciones ya que gracias a ellos he podido avanzar más rápido.
- Mi hermana, por hacer las pruebas de caja negra conmigo para ver que todo funcionaba correctamente.
- Sobre todo a mi hermana que se pico porque quería estar en los agradecimientos.
- A Soledad, mi pareja, que aportó ideas nuevas y fue la creadora del logo de la aplicación.
- A David Cotorruelo, por dedicarme algunas tarde para resolverme dudas que me surgían sobre el proyecto.
- A Daniel Muñiz, por su asesoramiento, así como en la resolución de las dudas que han ido surgiendo a lo largo del desarrollo del proyecto.



# ÍNDICE

<b>Introducción</b>	<b>4</b>
<b>Plan de Proyecto</b>	<b>5</b>
<b>Objetivos del proyecto</b>	<b>5</b>
<b>Plan de tareas</b>	<b>6</b>
<b>Diseño</b>	<b>7</b>
<b>Implementación</b>	<b>8</b>
<b>Presentación</b>	<b>9</b>
<b>Análisis</b>	<b>10</b>
<b>Entorno tecnológico</b>	<b>10</b>
<b>Especificación de Casos de Uso</b>	<b>10</b>
<b>Identificación y definición de Actores</b>	<b>10</b>
<b>Diagrama de casos de uso</b>	<b>11</b>
<b>Diseño</b>	<b>12</b>
<b>Diseño servidor</b>	<b>12</b>
<b>Diseño cliente</b>	<b>14</b>
<b>Diseño Base de datos</b>	<b>15</b>
<b>Implementación</b>	<b>16</b>
<b>Características técnicas</b>	<b>16</b>
<b>Manual de usuario</b>	<b>19</b>
<b>Servidor</b>	<b>19</b>
<b>Cliente</b>	<b>20</b>
<b>Conclusiones</b>	<b>23</b>
<b>Referencias</b>	<b>23</b>



# 1. Introducción

La idea que proyecto viene por un problema que me surgió al pasar unos documentos en formato PDF al ordenador, entonces me vi con la necesidad de usar WhatsApp o el correo que aunque rápidos y cómodos era un poco tedioso.

En WhatsApp debía tener un grupo conmigo para no molestar a nadie, pasarlo, descargarlo y luego guardarlo donde yo quería.

Con el correo es casi lo mismo, pero enviándome a mí el correo la única diferencia era la aplicación que usaba con lo que se me ocurrió un programa que no ocupase mucho y que solucionara este problema.

Mientras construía la aplicación era bastante común encontrar enlaces que podría servirme para el desarrollo del programa, por tanto pensé en agregar una nueva funcionalidad, el envío de una URL y que esta se abriera en el buscador por defecto del usuario.

En realidad la aplicación es bastante sencilla, es un servidor en el ordenador y una aplicación móvil que envía y recibe información del servidor. Para enviarla, lo hará mediante el protocolo TCP [1] y para recibirla usará el protocolo UDP [2].

Asimismo, la aplicación está diseñada para subir a “**Firebase**” [3] cierta información sobre el cliente, como las URLs que ha pasado a través de la misma, el nombre y tamaño de cada archivo.



## 2. Plan de Proyecto

### 2.1. Objetivos del proyecto

El objetivo del proyecto es la realización de una aplicación basada en Java que permita enviar información del móvil al ordenador de una forma cómoda y rápida.

1. **Enviar información al ordenador**
2. **Recibir información del móvil y gestionarlo**

De este modo, los objetivos del proyecto de manera detallada son los siguientes:

<b>Objetivo 1</b>	<b>Envio de informacion</b>
<b>Descripción</b>	Enviar información al ordenador de forma rápida
<b>Importancia</b>	Alta.
<b>Comentarios</b>	Enviar información a la ruta que nos haya indicado el servidor

<b>Objetivo 2</b>	<b>Recibir información</b>
<b>Descripción</b>	Recibir información del teléfono de forma rápida
<b>Importancia</b>	Alta.
<b>Comentarios</b>	Recibir información y escribirla en la ruta que tiene el servidor marcada



### 3. Plan de tareas

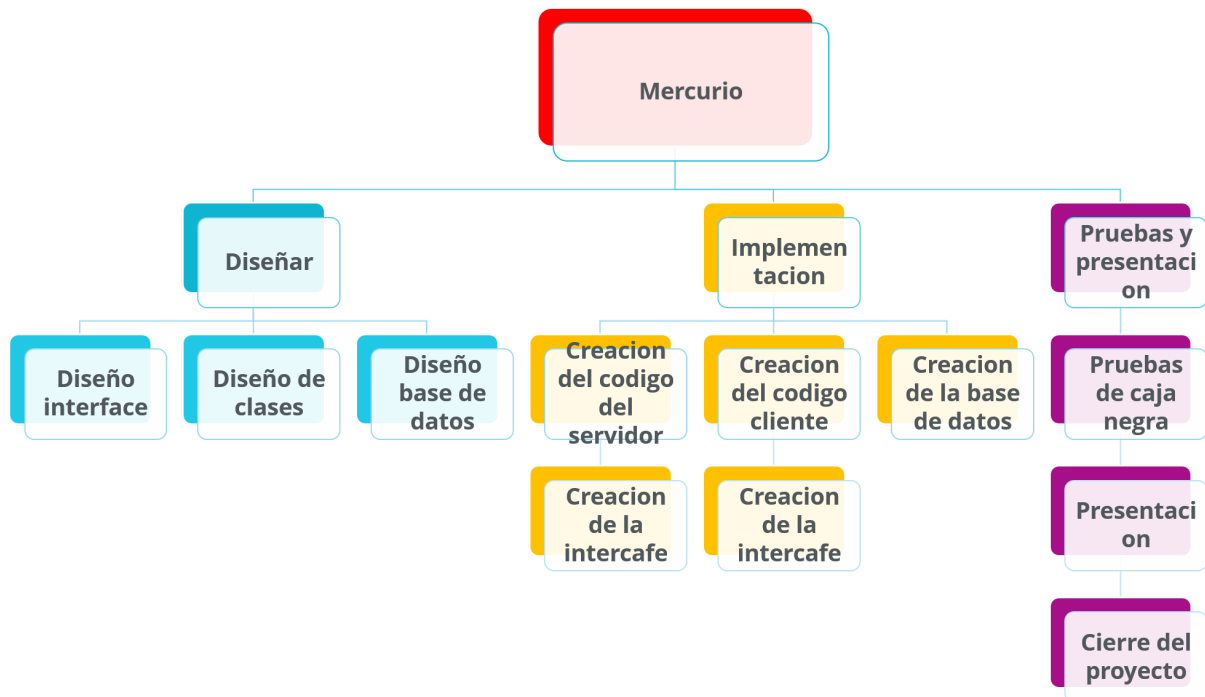


DIAGRAMA EDT

Como podemos observar, vemos las diferentes tareas que son necesarias para un buen desarrollo del proyecto. En primer lugar tenemos que diseñar la interfaz, el diseño de clases y la base de datos, cuando esto esté terminado debemos empezar con la implementación.

Comenzamos por programar el servidor con su interfaz y un cliente móvil que pueda conectarse con el servidor. Una vez hechas las pruebas de conexión y transferencias de archivos nos ponemos a crear la base de datos.

Por último hacemos las pruebas de caja negra y con esto ya se finaliza el proyecto.



### 3.1. Diseño

Código	Nombre	Responsable
3.1.1	Diseño interfaz	Miguel Chiara
Descripción: Se crea el diseño que tendrá la interface		

Código	Nombre	Responsable
3.1.2	Diseño de clases	Miguel Chiara
Descripción: Se crea el diseño que tendrán las clases para un correcto funcionamiento		

Código	Nombre	Responsable
3.1.3	Diseño de base de datos	Miguel Chiara
Descripción: Se crea el diseño de la base de datos para tener toda de la forma más clara y accesible		



## 3.2. Implementación

Código	Nombre	Responsable
3.2.1	Creación del servidor	Miguel Chiara
Descripción: Crearemos el servidor para conectarnos con la aplicación		

Código	Nombre	Responsable
3.2.2	Creacion del cliente	Miguel Chiara
Descripción: Crearemos el cliente para conectarse con el servidor junto con la base de datos		

Código	Nombre	Responsable
3.2.1	Control de calidad	Miguel Chiara
Descripción: Comprobamos que todo funciona perfectamente		





### 3.3. Presentación

Código	Nombre	Responsable
3.3.1	Pruebas de caja negra	Miguel Chiara
Descripción: En esta tarea, se deberá hacer las pruebas pertinentes para comprobar la robustez de la aplicación		

Código	Nombre	Responsable
3.3.2	Presentación	Miguel Chiara
Descripción: En esta tarea se deberá elaborar la presentación powerpoint, así como la preparación de la exposición oral.		

Código	Nombre	Responsable
3.3.3	Cierre del proyecto	Miguel Chiara
Descripción: Última tarea en la que se preparará el contenido con la solución final solicitado por la escuela, así como entregarlo en fecha.		



## 4. Análisis

### 4.1. Entorno tecnológico

La aplicación fue desarrollada en java 7 [4] y android 4.1 [5]

Por otro lado, para desarrollar la aplicación se necesitarán las siguientes herramientas:

- Netbeans 7.2 [6], que será utilizado para desarrollar el servidor en java.
- Android 4.1 [5] que será utilizado para desarrollar el cliente
- JDK 7 [4], que no es más que un conjunto de herramientas (programas y librerías) que necesarias para desarrollar (compilar, ejecutar, generar documentación, etc.) en lenguaje Java.
- Firebase [3] será la base de datos que usaré para guardar los datos de los usuarios de la aplicación.
- Lucidchart [7], para realizar diagramas, como pueden ser, por ejemplo, los casos de uso.

## 5. Especificación de Casos de Uso

### 5.1. Identificación y definición de Actores

Los actores identificados en este proyecto son los usuarios que harán uso de la aplicación una vez sea desarrollada y puesta en marcha.

Actores	Descripción
<b>ACTOR01: Usuario.</b>	Se trata del único y actor principal de la aplicación, que será el que ejecutará la aplicación y hará uso de ella.



## 5.2. Diagrama de casos de uso

A continuación vemos con detalle el diagrama de casos de uso de esta solución, donde se aprecian todas las funcionalidades que ofrece.

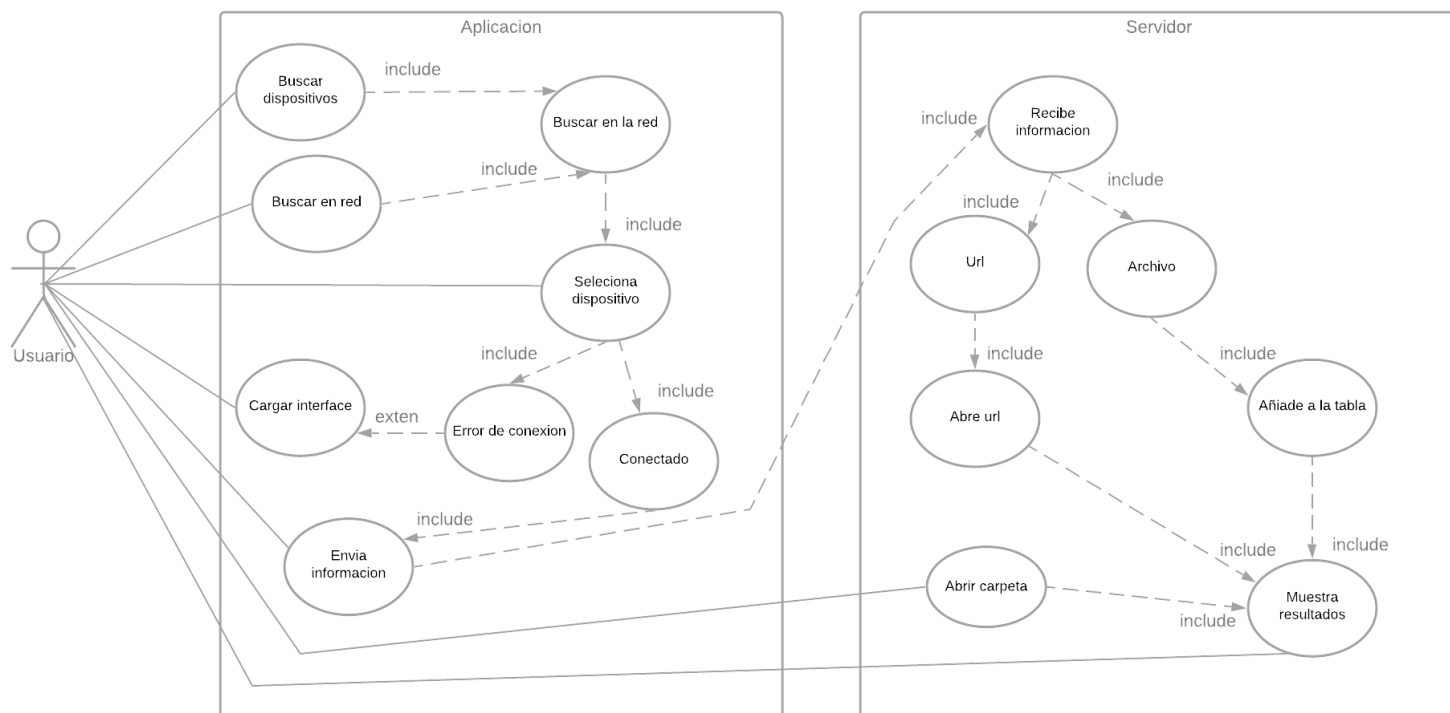


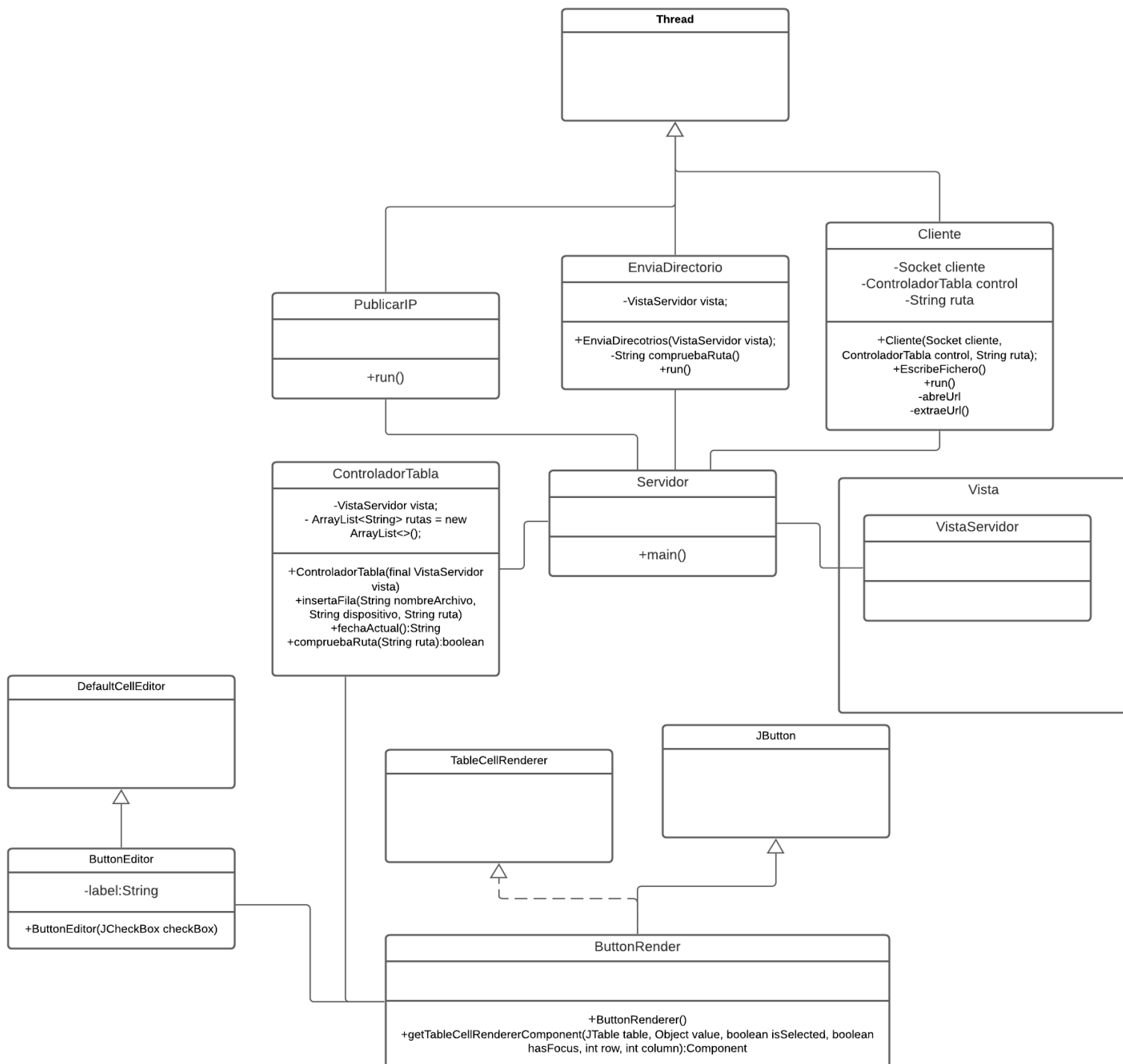
DIAGRAMA DE CASOS DE USO



# 6. Diseño

## 6.1. Diseño servidor

A continuación vemos con detalle el diseño correspondiente a la parte del servidor:



Para el diseño del servidor lo que hacemos es un hilo por archivo, ya que lo enfocamos a una red local y queremos que vaya lo más rápido posible un hilo por archivo permite pasar varios archivos a la vez.

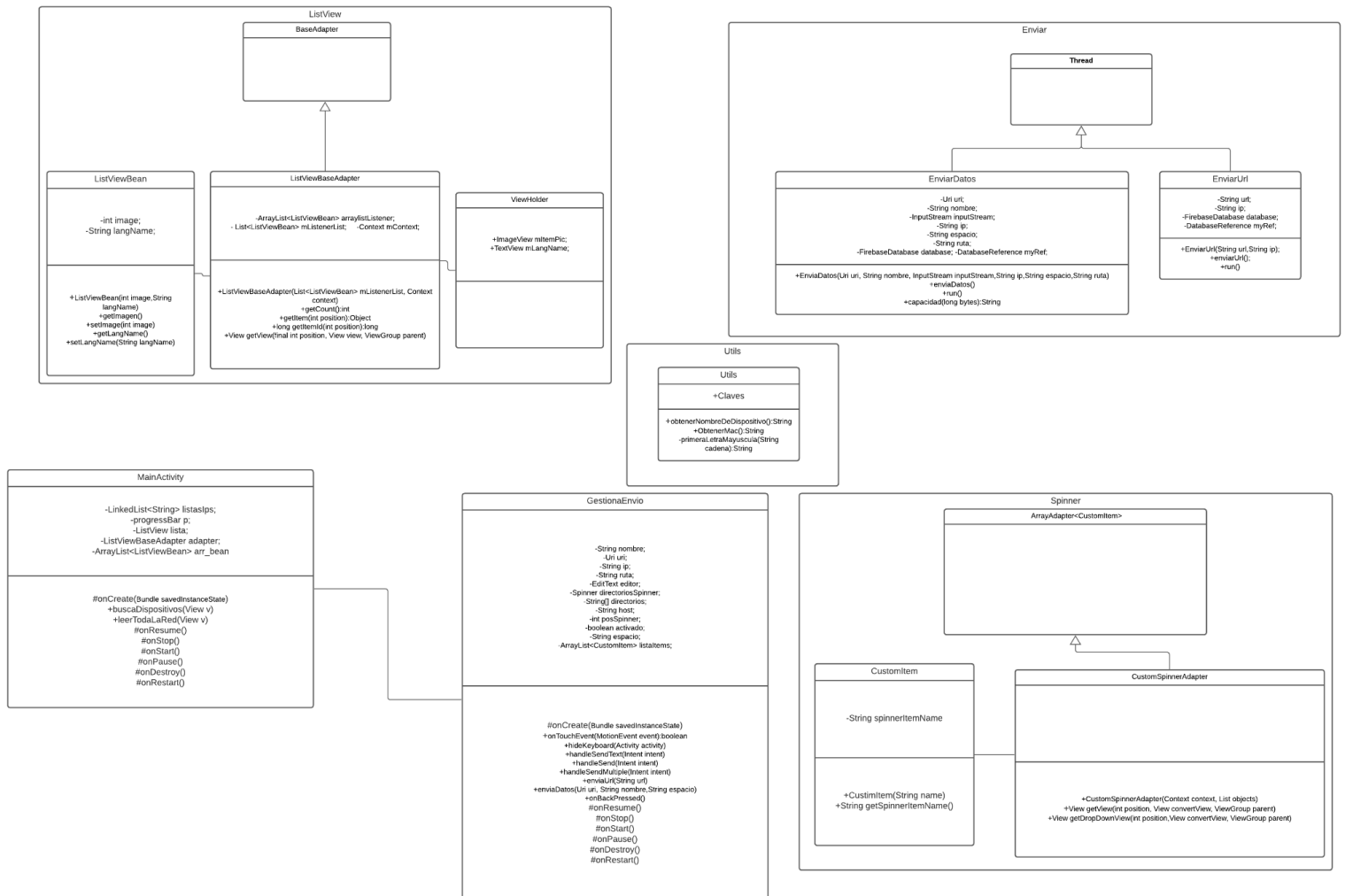
Cuando iniciamos el servidor este crea un grupo de red en el que manda un paquete de datos con el nombre del equipo una vez consigue establecer conexión con el cliente genera una conexión UDP con el cliente donde enviará los nombres de las carpetas que tiene disponible en la ruta especificada.

Acto seguido una vez el cliente actúe el servidor gestiona la información que le llegue en el caso de ser una URL abrirá una pestaña del explorador y en el caso de ser un archivo creará los archivos en la ruta que le haya indicado el cliente donde se guardará todo lo que le llegue.



## 6.2. Diseño cliente

A continuación vemos con detalle el diseño correspondiente a la parte del cliente:



Una vez inicia la aplicación tienes 2 botones uno que entra en el grupo de red que ha creado el servidor y otro que escanea la red entera y te dice los dispositivos disponibles en tu red.

una vez seleccionemos el dispositivo comprobará si el servidor esta abierto y en caso de que lo esté creará una conexión para la transferencia de archivos,

la transferencia de datos se hará mediante un intent que obtiene la uri del archivo o url y una vez obtenga la uri intentará establecer la conexión para comprobar que no se ha cerrado si lo consigue transferirá el archivo al servidor con los datos de ruta que el cliente desee.



### 6.3. Diseño Base de datos

A continuación vemos con detalle el diseño correspondiente a la parte la base de datos:



La base de datos agrega el dispositivo por mac, una vez el usuario lanza una url o un archivo se agregara en su respectiva tabla.



# 7. Implementación

## 7.1. Características técnicas

A continuación mostraremos la arquitectura técnica del sistema, donde podrá verse reflejado las necesidades previstas de almacenamiento, procesamiento y comunicaciones, así como las conexiones existentes entre ellas.

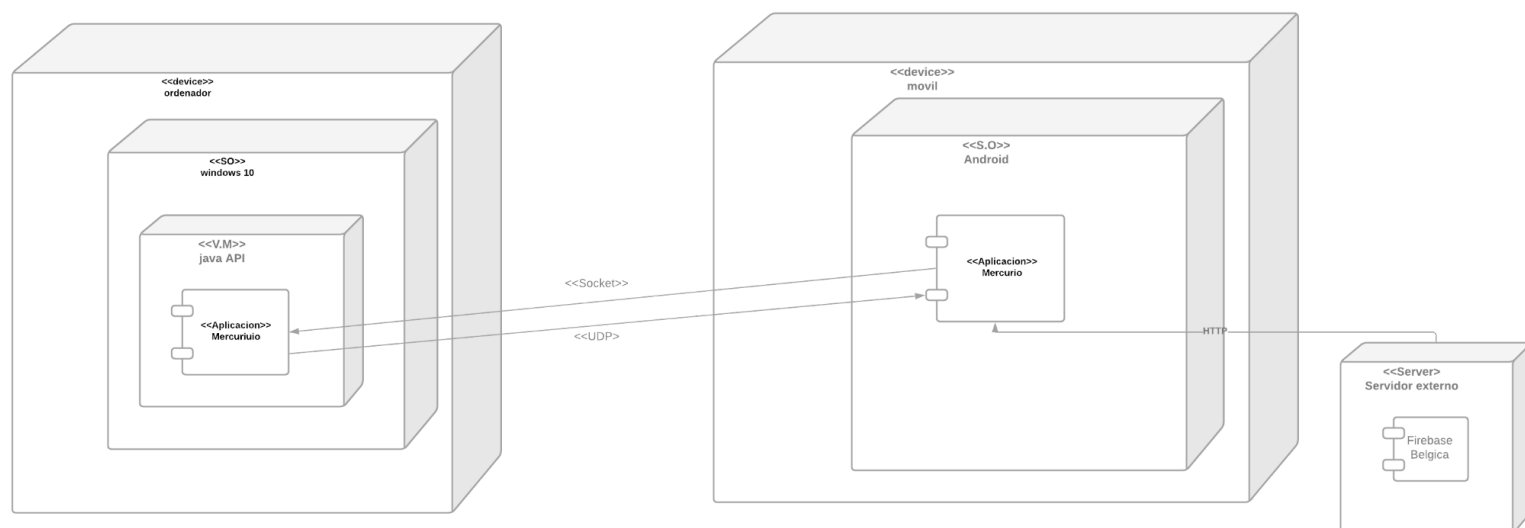


DIAGRAMA DE ARQUITECTURA TÉCNICA

Podemos observar en la imagen anterior que tenemos un sistema en un equipo con sistema operativo Windows 10 y otro con un sistema operativo en android.

El sistema tiene instalado un la Máquina Virtual de Java para el correcto funcionamiento de la aplicación, ya que dicha aplicación está realizada en el lenguaje de programación JSwing.

Esta aplicación hace de servidor permitiendo la conexión de varios dispositivos a la vez.

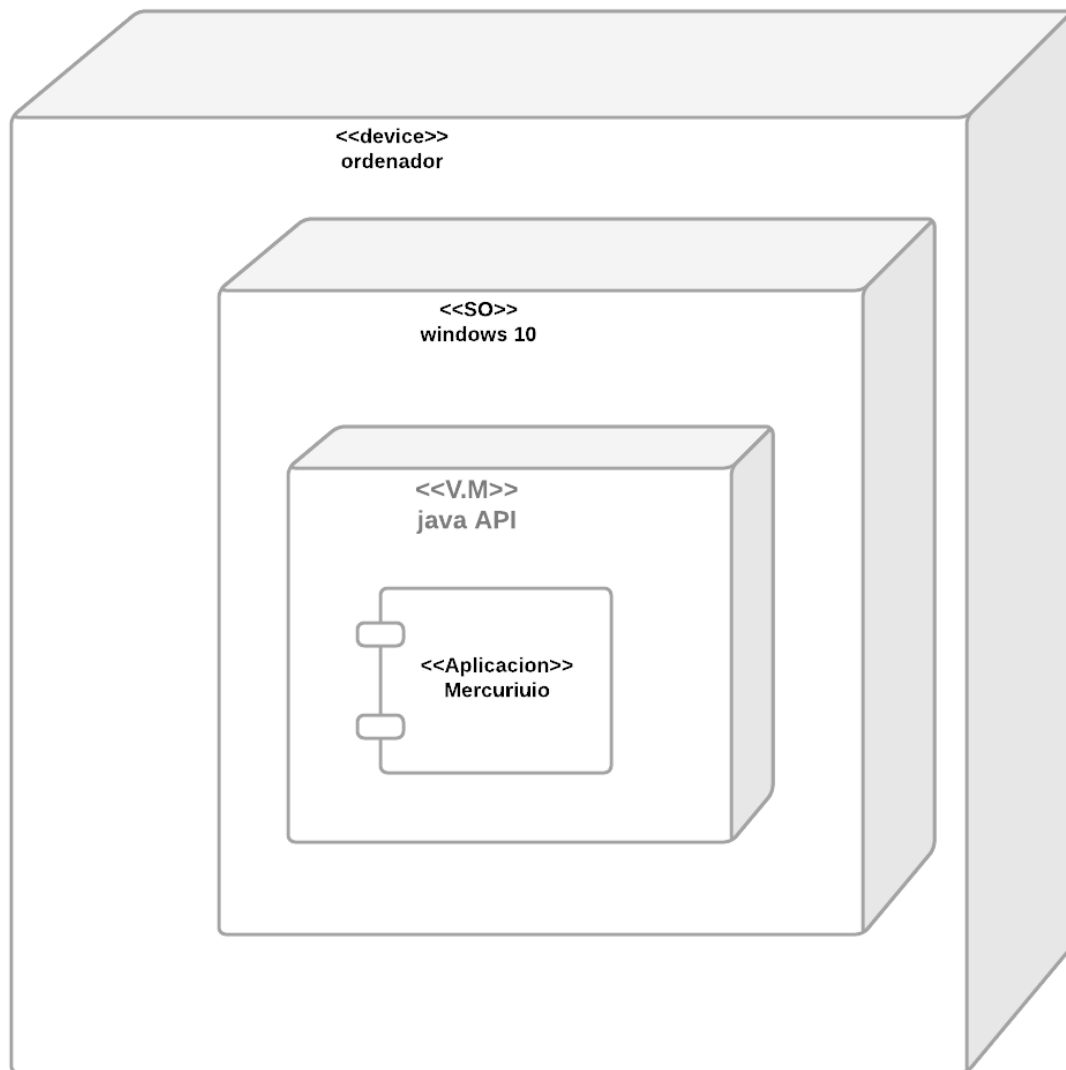
El sistema android es una aplicación en la que puedes enviar al servidor cualquier tipo de información y que se conecta a una base de datos para tener un control de lo que hace cada usuario.

Como se puede observar es una conexión TCP [1] para el traspaso de información del cliente al servidor y una conexión UDP [2] para la información que necesita el cliente para una transmisión de datos segura y fiable.





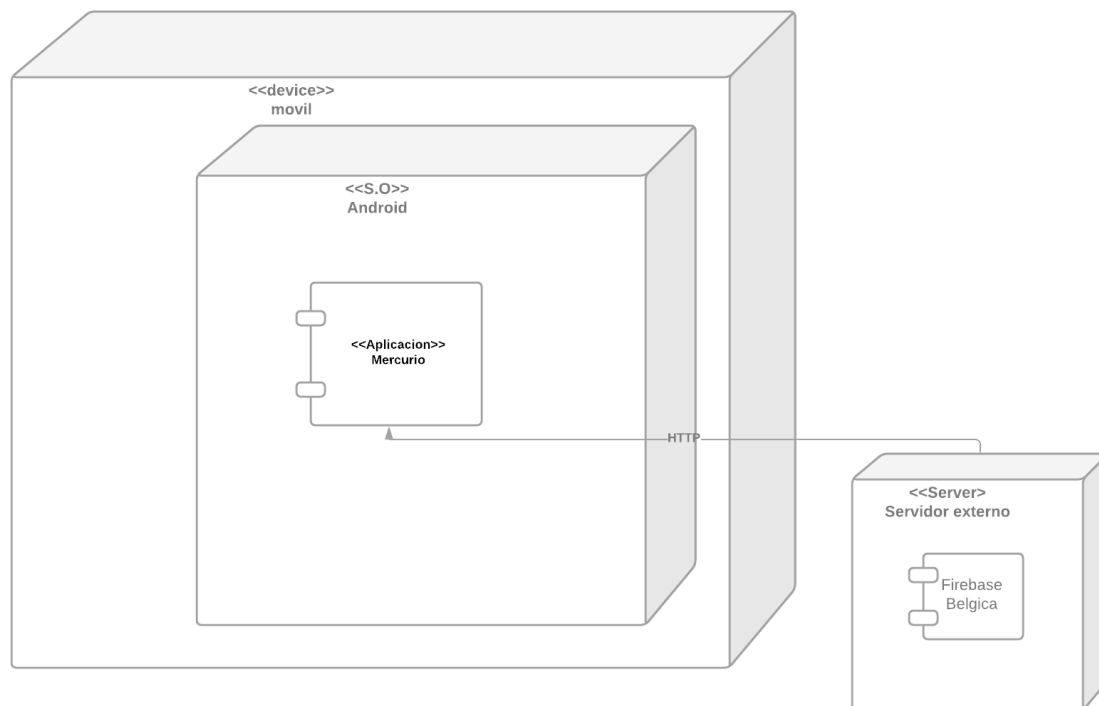
A Continuación vemos las capas del servidor:



**DIAGRAMA DE ARQUITECTURA DEL SERVIDOR**



A Continuación vemos las capas del cliente:



**DIAGRAMA DE ARQUITECTURA DEL CLIENTE**

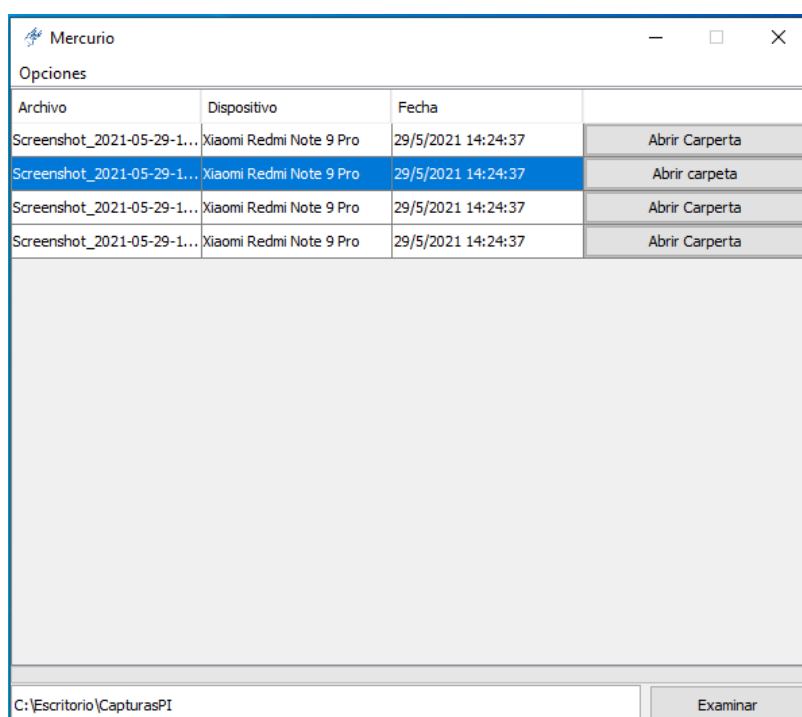


## 8. Manual de usuario

A continuación se desarrollará un breve manual de usuario, donde se explicará el habitual comportamiento de la aplicación. Al tratarse de una aplicación java, basta con ejecutarla para poder comenzar a utilizarla

### 8.1. Servidor

En primer lugar disponemos de la interfaz principal, tal y como vemos en la siguiente ilustración.



INTERFAZ - SERVIDOR

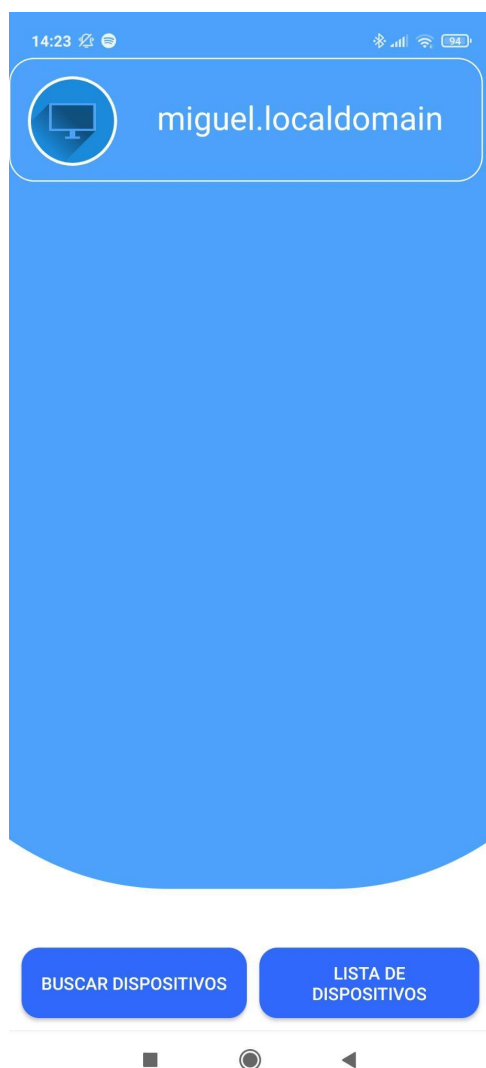
El funcionamiento del servidor es muy simple es una tabla donde irán llegando los archivos y tendremos algunos datos de estos como son el nombre la fecha y dispositivo desde el que llegó, en la parte inferior tenemos un cuadro editable que te permite seleccionar el directorio donde quieres que se guarde todo lo que llegue y por último un botón que te abre un explorador de archivos

También tienes un icono de escritorio que te permite ocultar la interfaz, mostrar la interfaz o cerrar la aplicación.



## 8.2. Cliente

En primer lugar disponemos de la interfaz principal, tal y como vemos en la siguiente ilustración.

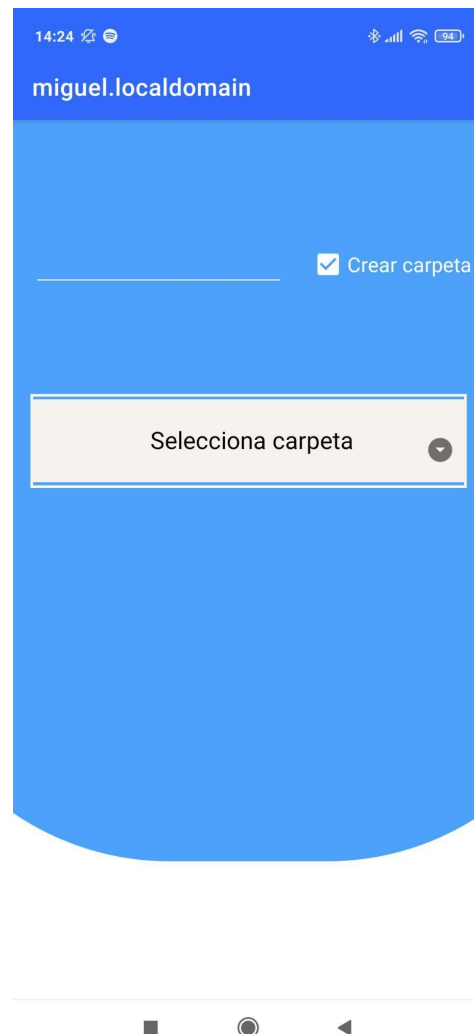


INTERFAZ 1 - CLIENTE

Como podemos ver tenemos 2 botones uno que nos permite buscar qué dispositivos tiene el servidor abierto y otro en caso de no encontrarlo que te muestra todos los dispositivos conectados.

Una vez tengamos los dispositivos seleccionamos el que queramos accediendo a la segunda interfaz.

En segundo lugar disponemos de la interfaz secundaria, tal y como vemos en la siguiente ilustración.



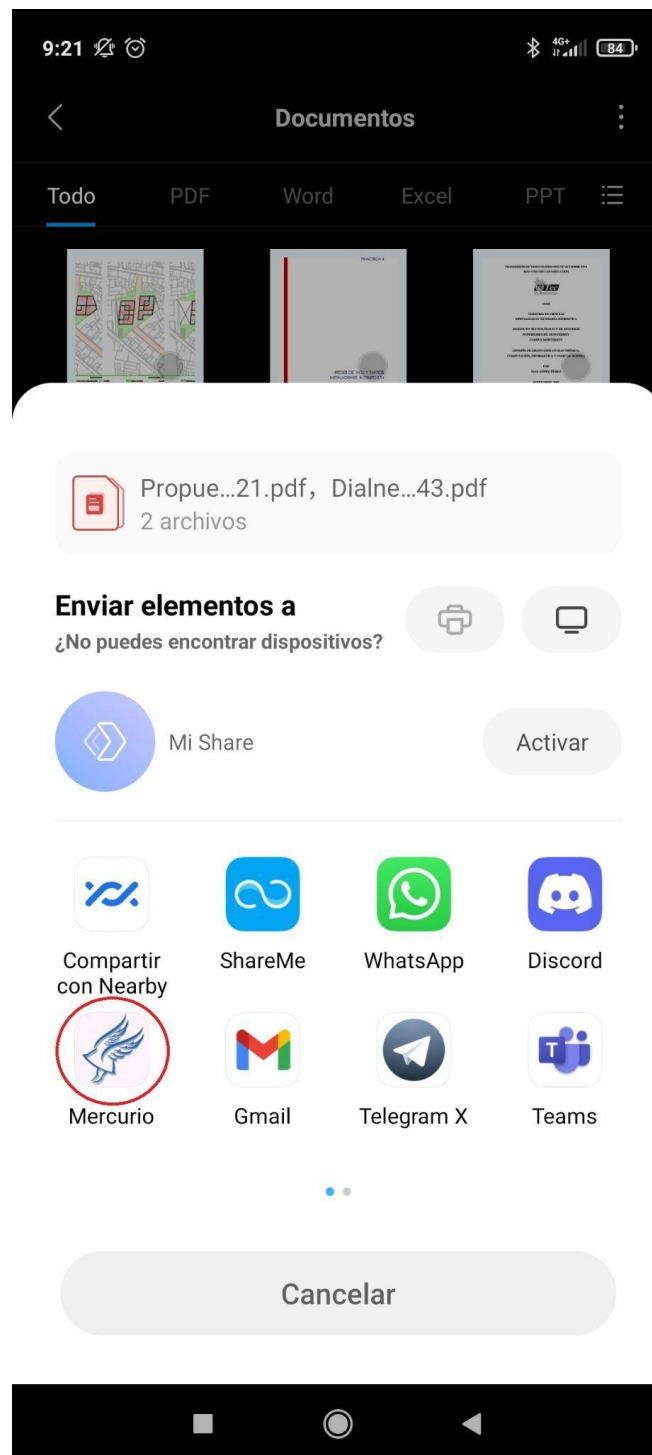
**INTERFAZ 2 - CLIENTE**

En esta interfaz tenemos 3 elementos

- Crear carpeta en el caso de estar seleccionado se habilitará el cuadro de diálogo donde podremos escribir el nombre de la carpeta, por el contrario si no está activo se deshabilita el cuadro de diálogo y se activa el spinner.
- Cuadro de diálogo en el caso de estar activo nos permite crear una carpeta en la ruta establecida en el servidor
- Spinner en el caso de estar activo veremos todas las carpetas de la ruta especificada en el servidor



Una vez esté todo con la configuración que tu quieres sales de la aplicación, seleccionas los archivos que quieras le das a compartir y te aparecerá la aplicación Mercurio



ICONO COMPARTIR - CLIENTE

Una vez le demos empezará a pasarse todo al ordenador justo donde le habíamos dicho

## 9. Conclusiones

Con esto, podemos concluir que, aunque ya existen soluciones viables a mi propuesta, lo que yo propongo logra hacerlo de una forma inmediata y cómoda, sin tener que hacer casi nada, tan solo abrir el servidor, conectarse y listo.

También me quedo con las enseñanzas adquiridas durante la realización de este proyecto, tanto en la transmisión de archivos como en conexiones, ya que la aplicación está basada, principalmente, en la conexión entre un móvil y un ordenador.

## 10. Referencias

[1] Protocolo de control de transmisión TCP. Es uno de los protocolos fundamentales en Internet. <https://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/tcp.html>

[2] Protocolo de control de transmisión UDP. Otro de los protocolos fundamentales en internet. <https://neo.lcc.uma.es/evirtual/cdd/tutorial/transporte/udp.html>

[3] Firebase es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles. <https://firebase.google.com>

[4] Java 7 es un lenguaje de programación y una plataforma informática. <https://docs.oracle.com/javase/7/docs/api/>

[5] Android es un sistema operativo móvil basado en núcleo Linux y otros software de código abierto. [https://www.android.com/intl/es\\_es/](https://www.android.com/intl/es_es/)

[6] NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

<https://netbeans.apache.org/>

[7] Lucidchart es una herramienta de diagramación basada en la web, que permite a los usuarios colaborar y trabajar juntos en tiempo real, creando diagramas de flujo, organigramas, esquemas de sitios web, diseños UML.

<https://www.lucidchart.com/>

