

Reporte de Laboratorio

Aplicación Capturador de Fotos Instantaneas Programadas para Elementos de la Naturaleza.

Rodrigo de la Garza Arredondo
Gerardo Loperena Bustillos
Pablo Alberto Ramírez Ramírez
Christian Alejandro Saldaña Calderón
Ingeniería en Tecnologías de la Información
Universidad Politécnica de Victoria

Resumen— En la elaboración de una aplicación móvil, al igual que en algún otro proyecto, es necesario llevar a cabo una planeación de desarrollo, en donde serán planteadas las ideas principales para solucionar la problemática a resolver, y en base a ello deberá ser creada la manera experimental del proyecto. A lo largo del documento se podrá encontrar la descripción puntual del trabajo, la descripción de desarrollo de esta aplicación, y los resultados obtenidos de la misma. Dando introducción al proyecto, la problemática planteada es desarrollar una aplicación móvil capaz de capturar fotos instantaneas de la naturaleza, con intervalos de tiempo definidos. Serán mencionados los puntos importantes que sirvieron para creación de esta aplicación, la manera de trabajo, los puntos de desarrollo por la parte de aplicación móvil, servidor que tomará la función como almacen de las fotos recolectadas y un sistema capaz de visualizar los resultados obtenidos. Para concluir este reporte de laboratorio, como fue mencionado, se podrán visualizar los resultados obtenidos de la misma.

I. INTRODUCCIÓN

Hoy en día, el uso de cámaras, celulares, tablets, entre muchos otros dispositivos, permiten la captura de fotografías, ya sea de manera directa, con temporizador, con iluminación incluida, enfoque para larga distancia, entre otras funcionalidades. El proyecto a desarrollar es un capturador de fotografías instantaneas de la naturaleza, siendo programado por el usuario definiendo intervalos de un tiempo estimado, la aplicación podrá seguir trabajando en segundo plano mientras el usuario realiza otras funciones, así como también se le será notificado las fotografías que haya realizado.

Por otro lado a mencionar, se busca que las imagenes que la aplicación pueda tomar sean almacenadas y visualizadas en un servidor web, creando así carpetas o apartados donde estas mismas serán almacenadas. Añadiendo a la parte del desarrollo por medio de un script se espera poder realizar y visualizar lo mencionado previamente, cumpliendo con tres funcionalidades; Poder programar de manera manual la captura de fotografías, realizar la programación de la misma aplicación de manera automática ingresando la fecha en la que la petición es realizada asignando los intervalos de tiempo, y finalmente poder visualizar los apartados, como las fotografías, realizar una descarga de estos archivos o por otra parte, eliminarlos.

Con la ayuda de Android Studio [1], la cuál es una herramienta para la creación de aplicaciones móviles utilizando el lenguaje de programación Java[2], se busca llevar a cabo la realización de este proyecto.[3] Además, haciendo uso del servidor 000webhost[4], se busca que sirva de almacenamiento para las fotografías capturadas, brindando al usuario un lugar donde poder visualizarlas y almacenarlo dentro de una base de datos. Por último punto a mencionar, con ayuda del lenguaje python [5] se busca la realización del script capaz de programar, visualizar, descargar y eliminar los proyectos que el usuario genere utilizando la aplicación.

II. DESARROLLO EXPERIMENTAL

Dentro del plan para desarrollar la aplicación web, ha sido necesario aplicar los conocimientos obtenidos a lo largo del curso, en base a las prácticas realizadas y el conocimiento adquiriendo del lenguaje de programación Java.[6]

Desarrollo de Aplicación Móvil.

Dando comienzo a esto proyecto, por el lado de la aplicación en su funcionalidad cuando el usuario ingresa a la aplicación y presiona el botón "Preparado", la aplicación entra en un ciclo while, en espera de que nuevos parámetros sean ingresados a la base de datos desde el script de python. Cabe resaltar, esta solicitud se realiza cada 5 segundos para evitar saturar la aplicación en cuanto a peticiones a la base de datos.

La aplicación se conecta a la base de datos mediante un AsyncTask, el cual realiza las peticiones a la base de datos en segundo plano (doInBackground) y guarda dicha solicitud en sus respectivas variables.

Cuando se detecta la entrada de nuevos parámetros (Es decir, una orden de capturar fotografías), los almacena y en diferentes TextView se muestra la información indispensable, como; La cantidad de fotografías que se requieren tomar, en cuanto tiempo finalizará la captura de fotografías, en cuando se iniciará la captura de fotos, la cantidad de fotografías que se tomarán y el intervalo de segundos que habrá entre la

captura de cada foto y como se van comportando conforme la aplicación se ejecuta.

Después, en otro proceso se evalúa si la fecha de inicio de la orden es anterior a la fecha actual. De ser así, la orden se ejecutará de manera instantánea. De lo contrario, se manda a llamar un temporizador (CountDownTimer) en horas, minutos y segundos del tiempo que indica cuanto tiempo falta para que la orden se pueda ejecutar. Una vez este contador llega a cero, empieza con la orden de captura de fotografías.

De igual forma, se cambia el TextView el cual muestra el estado de la aplicación y coloca un texto indicándole que la orden está en proceso.

En el instante que la aplicación comienza su función se inicia el temporizador de tiempo restante, el cual es calculado mediante la multiplicación del intervalo (en segundos) de cada foto por la cantidad de fotos que se van a tomar. Luego se inicia el temporizador del intervalo de segundos entre capturas de fotografías. Cuando este contador finaliza se procede a tomar la fotografía, guardando así la ruta de la imagen y la imagen codificada (Ambos de tipo String) en un arreglo y cada que se toma una fotografía, la cantidad de fotos restantes disminuye. Ese temporizador se vuelve a repetir si es que aun quedan fotos pendientes por tomar. Cuando ya no quedan imágenes pendientes, la aplicación procede a subir las imágenes al servidor utilizando un AsyncTask. Se evalúa la cantidad de imágenes que se guardaron en el Array de rutas de imágenes para llamar al método subirFoto, el cual es el encargado de llamar el AsyncTask para subir la fotografía. Sin antes mandarle a dicho método la ruta de la imagen y la imagen codificada.

Pantalla Principal.

En una primera visualización, se pueden encontrar con dos elementos indispensables, aplicando las herramientas brindadas por Android Studio podemos encontrar dos RadioButtons.

Uno permite al usuario seleccionar alguno de los proyectos que haya creado previamente. En caso de no haber proyectos, el radio button es deshabilitado.

Mientras que el otro, permite al usuario crear un nuevo proyecto, habilitando un EditText en el cual el usuario puede ingresar el nombre del proyecto que desea agregar. En caso de que el proyecto ya exista, las fotografías tomadas serán colocados en el proyecto ya existente. En caso contrario, crea en la base de datos el nuevo proyecto.

Una vez seleccionada alguna de las dos opciones, se habilita el botón que permite al usuario mostrarle la interfaz en la cual se podrán capturar las fotografías.

Pantalla Captura de Fotografías.

En esta pantalla se muestra la vista previa de la cámara (Siempre que el usuario le haya concedido los permisos necesarios a la aplicación).

Esta pantalla contiene como elementos:

- El nombre del proyecto al cual se van a capturar los datos.
- Un botón el cual le permite al usuario reiniciar la aplicación para que vuelva a capturar fotografías.
- La vista previa de su cámara trasera (Siempre y cuando el usuario le haya concedido los permisos necesarios a la aplicación).
- Un TextView el cual muestra el estado Actual de la aplicación. Este cambiará cuando el botón sea pulsado y la orden para iniciar la captura sea mandada desde el script.
- El botón "Modo Espera" pausa la aplicación, en espera de recibir la orden desde python.

Finalmente, como un punto adicional, una serie de TextViews diversa información como la cantidad de fotografías que quedan por tomar, en cuanto tiempo finalizará la captura de fotografías, en cuando se iniciará la captura de fotos, la cantidad de fotografías que se tomarán y el intervalo de segundos que habrá entre la captura de cada foto, como se mencionó en puntos anteriores.

Desarrollo de la Aplicación Web.

De manera principal, se crea una interfaz en la cual el usuario podrá acceder mediante el script de python. En esta página web el usuario podrá ver todos los proyectos que hayan sido creados y las respectivas fotografías de cada uno de estos proyectos.

Esta página web se encuentra alojada en el servidor 000webhost, en el cual también se creó la respectiva base de datos para el funcionamiento de la aplicación.

Al abrir cualquiera de los proyectos, el usuario podrá ver una a una todas las fotografías que ha capturado.

A su vez, el usuario podrá descargar el zip del proyecto el cual incluye todas sus fotografías. Se utilizó la clase ZipArchive() de PHP para poder realizar esta función, en la cual crea el zip con el nombre del proyecto que quiere descargar, dentro de este zip genera una carpeta con el mismo nombre que el zip, y dentro de ella carga todas las fotografías que se encuentren relacionadas con el id del proyecto seleccionado.

Por último, el usuario tendrá la posibilidad de borrar cualquiera de los proyectos en caso de que ya no sean necesarios para el.

Desarrollo de Script en Python.

Por el lado de desarrollo de python, en el servidor no se permitía realizar una conexión de la Base de datos con el sistema creado, las librerías no permitía conexiones directas [7], sin embargo, ayuda del método get para obtener el listado en el servidor podía obtenerse información de dicha página.[8] Otras de las funciones con las que contaba el programa era con cuatro opciones:

1. Programar fotos instantaneas de manera manual, ingresando los parámetros, como la fecha y hora con un

formato establecido, seleccionando el proyecto donde serían almacenadas las fotografías instantaneas y el intervalo de tiempo de cada fotografía.

2. Ingresar al servidor web creando una ventana emergente y realizar diferentes acciones como; Descarga, apertura y eliminación de las fotografías adquiridas, extrayendo los proyectos encontrados.
3. Tomar fotografías instantaneas programadas del tiempo de manera automática, agregando valores como el intervalo de tiempo, y el archivo donde serían registrados.
4. Terminar la ejecución del archivo.

Errores Encontrados a lo Largo del Desarrollo y Soluciones.

El primer error que fue encontrado es que al momento de realizar las peticiones a la base de datos desde la aplicación móvil. Se manejaron distintos archivos PHP para consultar los nombres de los proyectos y para consultar los parámetros, pero al momento de realizar múltiples peticiones al AsyncTask, esta solo realizaba la solicitud de uno de los URLs, por lo que capturaba los datos de la segunda página solicitada. Esto se debió a que, en Android, existe un límite al ejecutar múltiples hilos a la vez.[9] Por lo que, al momento de mandar a llamar al primer hilo, este se seguía ejecutando cuando se llamaba al segundo hilo. Por lo que el segundo hilo también obtenía los datos recibidos por el primero y de esta forma surgía el error. La solución fue juntar los resultados de ambas páginas PHP en uno solo, de esta forma solo fue necesario separar los datos mediante un carácter especial para posteriormente separarlos a como correspondía.

El siguiente error que se presentó fue que al momento de enviar imágenes al servidor la última foto capturada no se enviaba, y también cuando se solicitaba capturar una sola fotografía no la mandaba de forma correcta. Esto fue ocasionado debido a que el hilo que tomaba la fotografía era ejecutado después de que se intentara enviar dicha fotografía al servidor. Esto ocurría por que el hilo se mandaba a segundo plano (doInBackground) y no terminaba dicho proceso antes de que se procediera a subir la imagen al servidor. La solución fue transformar el hilo de la captura de fotografías a un método. Ya que de esta forma el proceso tenía que finalizar para proceder a subir la imagen al servidor, por lo que ya no se pasaba a segundo plano y el método subirFoto() era invocado cuando la foto ya era capturada.

El último error que se manifestó en el desarrollo del programa fue con el gestor de tiempos y fechas de java "java.time".[10] Este paquete solo está disponible para API 26 o superior, por lo que en dispositivos con android 8 o superior si funcionaba, pero en los que tenían una versión más antigua no. Este problema fue solucionado con la librería ThreeTen-Backport, la cual proporciona un backport de las clases de datetime de Java 8 para Java SE 6 y 7. Esta librería fue importada en el build.gradle de la aplicación. Luego solo se substituyó el import de las clases de java por las de Threetenbp

III. RESULTADOS

Finalmente, como resultado obtenido, se obtuvo la funcionalidad correcta de la aplicación móvil.

Al momento de ingresar, la primera pestaña, como se puede visualizar en la Figura 1, es la primera ejecución de la aplicación, iniciando con la creación de un nuevo proyecto, en donde serán almacenadas las fotografías capturadas por parte de la aplicación, para después almacenarlos en el servidor.

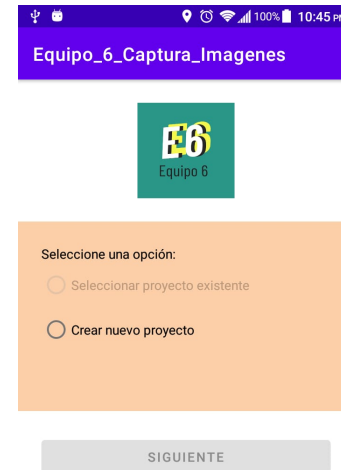


Figura 1: Primera ejecución de la aplicación

Una vez seleccionado la opción de crear un nuevo proyecto, se asigna nombre a la carpeta, como se puede ver en la Figura 2.

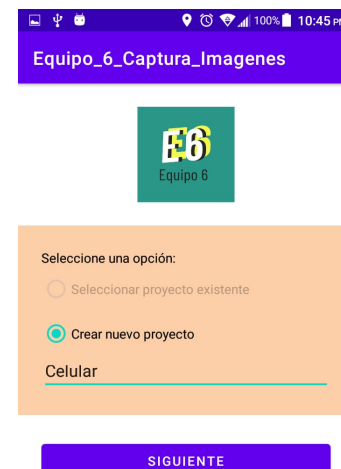


Figura 2: Asignando nombre del proyecto

Al momento de ingresar a la aplicación y capturar las fotografías, como se puede ver en la Figura 3 se da inicio a los intervalos de tiempo para comenzar a capturar fotografías.

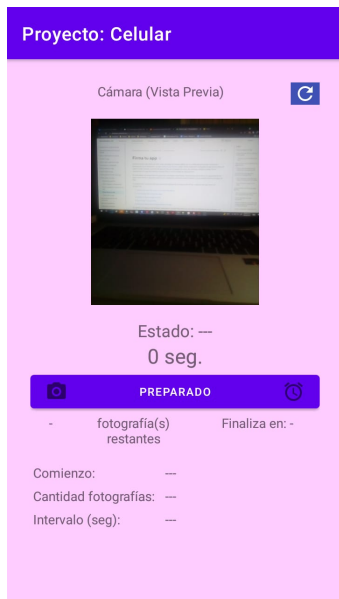


Figura 3: Iniciando captura de fotografías

Por otra parte en la Figura 4 se puede ver como la aplicación esta en momento de ejecución mostrando las fotografías restantes y el tiempo en el que finalizará.

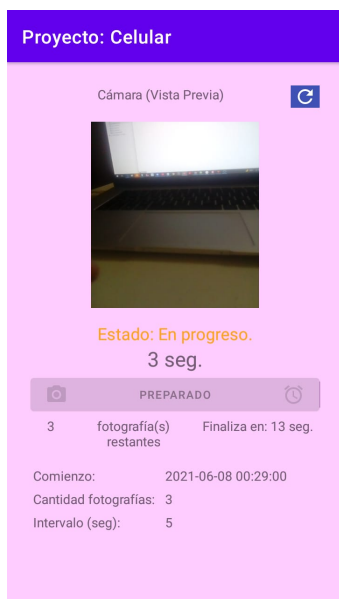


Figura 4: Estado en proceso

Una vez ejecutada la aplicación, en el momento final de la ejecución se almacena de manera local el proyecto creado, dando así la información de la ruta donde puede ser encontrado, se puede visualizar en la Figura 5.



Figura 5: Almacenamiento del proyecto

Una vez terminado el proceso de la captura de fotografías, en la Figura 6 se muestra la información obtenida final, del proceso realizado.

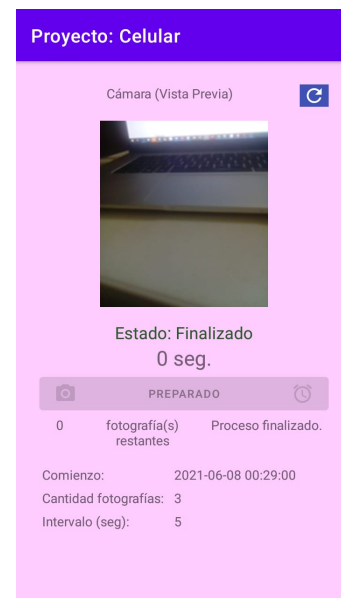


Figura 6: Estado Finalizado

Por otra parte, mostrando la segunda interacción del usuario en la Figura 7 el usuario es libre de seleccionar un proyecto previamente creado para así añadir nuevas fotografías que deseen ser almacenadas dentro del proyecto elegido.

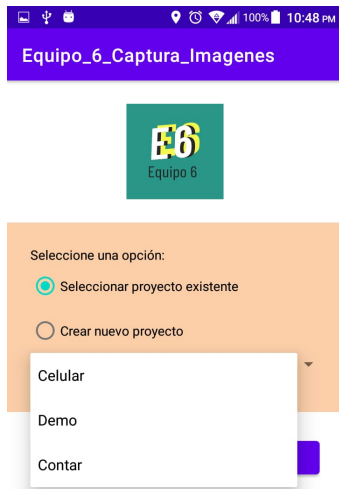


Figura 7: Proyecto Existente

En el lado del servidor podemos encontrar en la Figura 8 la visualización del proyecto almacenado con las capturas de la aplicación obtenidas.

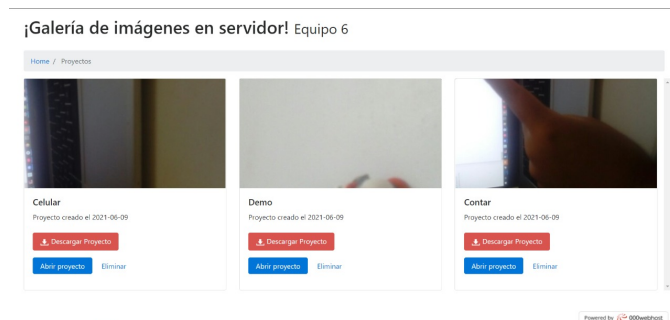


Figura 8: Galería Servidor Web

Para ingresar un proyecto en la Figura 9 tenemos como se visualiza estar dentro de la carpeta creada para almacenar diferentes fotografías.



Figura 9: Galería Proyecto

Al igual que de manera local en la aplicación, el servidor cuenta con la opción de descarga como .zip dentro la página para que el usuario sea libre de descargar desde cualquier lugar las fotografías capturadas. Se puede visualizar su ejecución en la Figura 10.

¡Galería de imágenes en servidor! Equipo 6

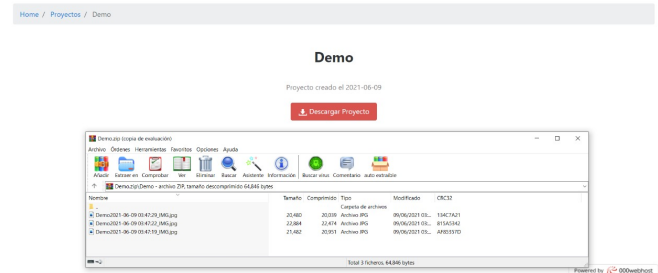


Figura 10: Descarga de Proyecto

Finalmente, todo esta gestión se almacena en la base de datos del servidor, que por la Figura 11 podemos visualizar la estructuración de las carpetas que son creadas al momento de ejecutar esta aplicación.

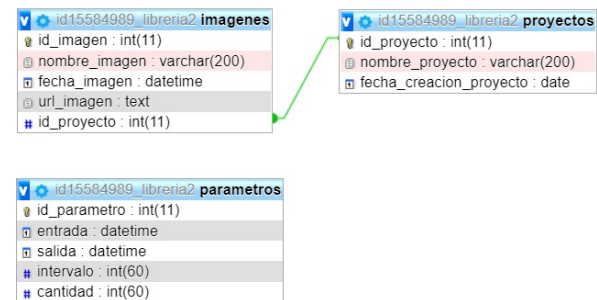


Figura 11: Base de Datos

REFERENCIAS

- [1] Adam Gerber; Clifton Craig. *Learn Android Studio Build Android Apps Quickly and Effectively*. Apress.
- [2] Jorge Martínez Ladrón de Guevara. *Fundamentos de Programación en Java*. Facultad Informática Universidad Complutense de Madrid.
- [3] Android Community Experts. *Android CookBook*. O'Reilly, 2011.
- [4] 000webhost. <https://www.000webhost.com/website-faq>. Consultado el 09-06-2021.
- [5] Ryan Mitchel. *Web Scraping with Python*. O'Reilly Media, Inc., 2018.
- [6] *Android documentation*. <https://developer.android.com/docs>. Consultado el 09-06-2021.
- [7] *Cómo controlar de navegador web*. <https://docs.python.org/es/3/library/webbrowser.html>. Consultado el 09-06-2021.

- [8] *Requests: HTTP para Humanos*. <https://docs.python-requests.org/es/latest/>. Consultado el 09-06-2021.
- [9] *Running Multiple Asynctasks at the Same Time Not Possible*. <https://stackoverflow.com/questions/4068984/running-multiple-asynctasks-at-the-same-time-not-possible>. Consultado el 09-06-2021.
- [10] *Introducción a Java Time*. <https://www.arteco-consulting.com/introduccion-a-java-time/>. Consultado el 09-06-2021.