

Universidad Autónoma de Yucatán Facultad de Ingeniería

Ingeniería Mecatrónica

Herramientas avanzadas de programación

Profesor: Ing. Edson G. Estrada López

Reporte del proyecto 2

2 Do List

Dzay Villanueva Luis

González Ávila Víctor Antonio

20/11/2018

Objetivo

Desarrollar una aplicación que sea capaz de agendar una lista de actividades pendientes y que permita una fácil edición de estas y que conserve las tareas a pesar de cerrar la aplicación.

Permitir una búsqueda y filtrado rápido de las actividades, para que el usuario tenga una mayor accesibilidad en la aplicación.

Lograr que el usuario tenga una gestión de las actividades ya concluidas o vencidas, como lo es la eliminación de estas tareas o el cambio de fecha.

Definición del proyecto

2DoList es una aplicación tipo agenda que permite la administración de tareas de manera sencilla, así como el fácil filtrado por las características que fueron añadidas a estas, su orden fácil, y la opción de modificar alguna tarea. Además, se presenta la opción de poder concluir una tarea o una eliminarla, las tareas concluidas podrán ser vistas en una pantalla especial.

Especificación de requerimientos

<i>Versión del O.S necesario.</i>	Android 4.0 o superior
<i>Espacio disponible en memoria</i>	Al menos 3 Mb

Tabla 1 Requerimientos

Alcance del proyecto

Se espera que *2DoList* sea una aplicación de fácil uso para el usuario, ya que se incorpora una interfaz intuitiva, con botones que facilitan el reconocimiento de su función.



Figura 1. Logotipo de la aplicación.

Herramientas utilizadas

A continuación, se presenta un listado de las herramientas gráficas usadas en la aplicación.

1. RecyclerView.
2. Toggle Button.
3. Button.
4. Scroll View
5. Fragments.
6. Edit Text
7. Switches
8. Spinners
9. Floating Action Button
10. Text View
11. Image View

Marco teórico

1. RecyclerView.

El RecyclerView es una versión más avanzada y flexible de ListView.

En el RecyclerView, varios componentes diferentes trabajan juntos para mostrar sus datos. El contenedor general de su interfaz de usuario es un RecyclerViewobjeto que agrega a su diseño. El RecyclerView llena de vistas proporcionadas por un administrador de diseño que usted proporciona.

2. Toggle Button.

Este es un tipo de botón que permite alternar una configuración entre dos estados, el estado nuevo se mantiene hasta que se presiona de nuevo el botón.

3. Button.

Este es un interruptor que lanza cierta operación al ser pulsado.

4. Scroll View

Este es un tipo de vista que nos permite el desplazamiento en orden jerárquico de objetos que se le fueron añadiendo.

5. Fragments.

Un fragmento es una sección modular de una actividad que tiene su ciclo de vida propio, recibe sus propios eventos de entrada y que puedes agregar o quitar mientras la actividad se esté ejecutando

6. Edit Text

Este es un elemento de la interfaz de usuario que permite la entrada y modificación de texto.

7. Switches

Este es un tipo de interruptor que permite alternar una configuración entre dos estados, el estado nuevo se mantiene hasta que se regresa el switch a su posición original.

8. Spinners

Esta es una lista que muestra un solo parámetro a la vez, los elementos de este vienen de un adapter asociado a él.

9. Floating Action Button

Este es un simple botón que flota en la pantalla, el usuario lo puede anclar en la parte que desee de la pantalla.

10. Text View

Este es un elemento de la interfaz gráfica que permite la visualización de texto.

11. Image View

Este es un elemento de la interfaz gráfica que permite la impresión y visualización de imágenes.

Descripción de la arquitectura del sistema

La aplicación está conformada por 4 Activities:

- Main Activity
- Segundo Activity (Agregar o modificar tarea)
- Tercer Activity (Filtrar tarea)
- Cuarta Activity (Ordenar tareas)

Empezando por describir el **Main Activity**, éste se conforma por la siguiente estructura:

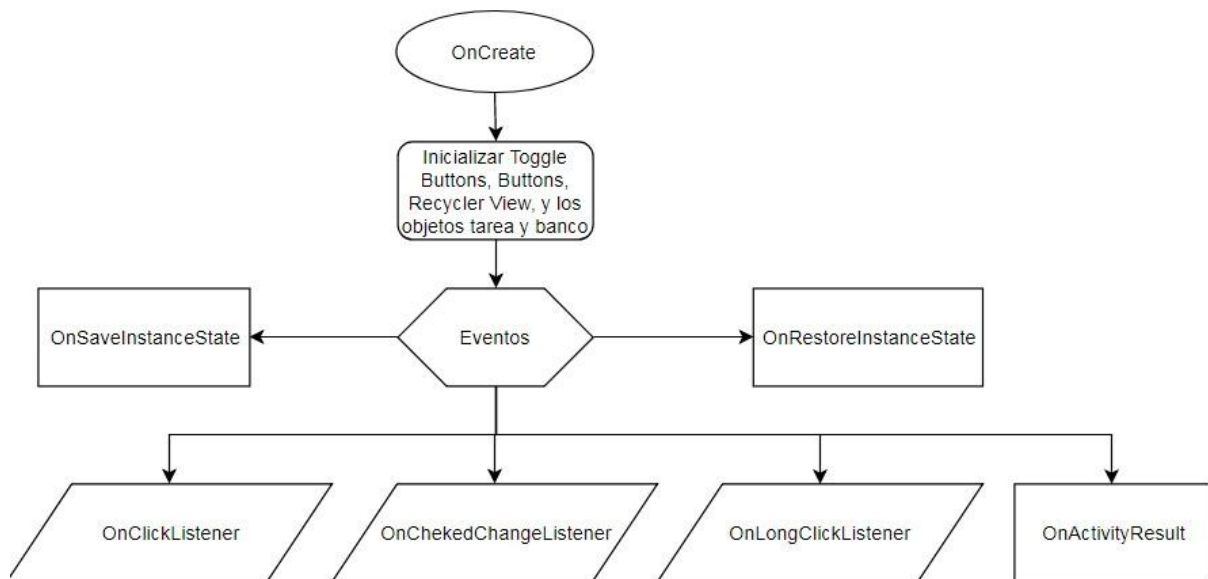


Figura 2. MainActivity.

Aunque haciendo un paréntesis, básicamente todas las Activities tienen una estructura similar.

El evento “OnClickListener” se basa en el clic a algún ítem del RecyclerView.

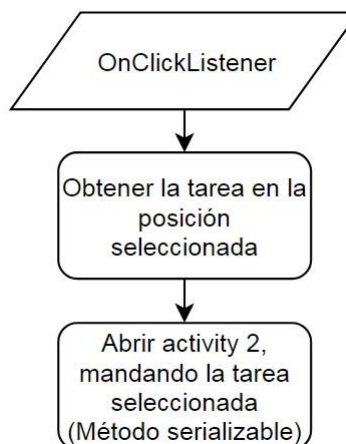


Figura 3. Evento OnClickListener.

Básicamente, como se indica en la imagen, lo que hace el evento es obtener la tarea en la posición del ítem del Recycler, para después mandarla al Activity 2, mediante el método “Serializable” (La clase tiene que tener “implements Serializable” para que se puedan enviar objetos entre Activities).

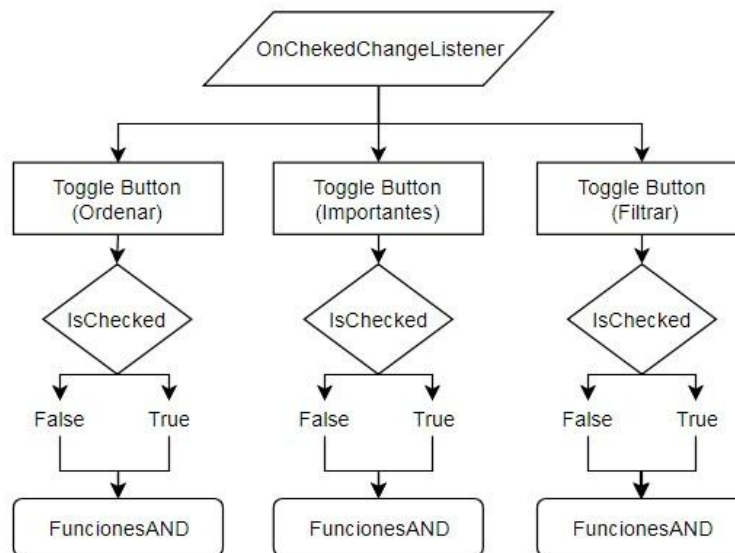


Figura 4. Evento `OnCheckedChangeListener`.

Como se puede observar, cada `Toggle Button` tiene la misma función independientemente de si se activó, o se desactivó. Esto es porque dentro de la función, lo que se hace es lo siguiente:

FuncionesAND: Lo que se hace, es verificar cuáles `Toggle Buttons` están activados, y obtener la combinación entre los 3 (“000”, “001”, “010”... “111”), para entrar a un switch, que haga la correspondiente combinación entre filtrado y ordenado.

En el evento `OnLongClickListener`, se agregó como extra, que al presionar el ítem en el `Recycler`, preguntara si se desea borrar la tarea o no.

En el evento `OnSaveInstanceState`, lo que se hace es almacenar básicamente el Banco de tareas, junto con sus variables necesarias para funcionar, ya que ahí se tiene toda la información de la aplicación. Y en el `OnRestoreInstanceState`, se recupera esa información.

En el evento `OnActivityResult`, lo que se hace es obtener las variables que fueron mandadas desde las demás `Activities` (cabe mencionar que todas envían un resultado, y que el `MainActivity` nunca se cierra).

En el **segundo Activity**, lo que se hace es lo siguiente.

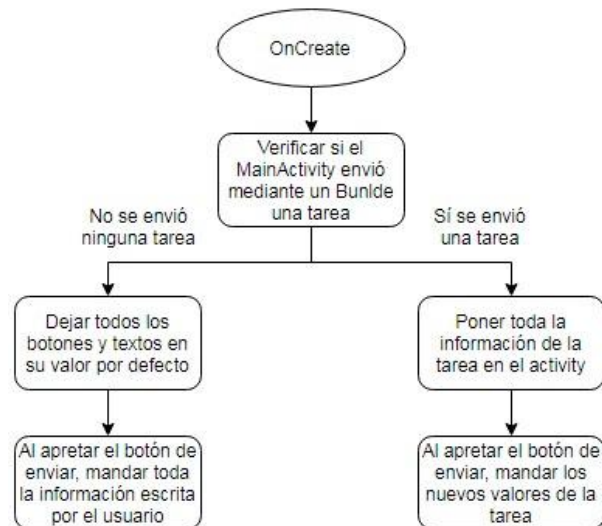


Figura 5. Segundo Activity.

Para cada caso, de si se envió una tarea o no, se manda un distinto resultado al MainActivity para el onActivityResult.

Para el caso de los **Activities 3 y 4** es básicamente la misma estructura que en el Activity 2, ya que **todos** los datos son recompilados de los Activities 2, 3 y 4, para procesarlos en el MainActivity, por lo que únicamente se envían y reciben los valores de los botones y textos.

En resumen, la aplicación se conforma por las siguientes clases:



Figura 6. Evento Clases de la aplicación.

La clase Tarea: tiene los siguientes argumentos: Título, Descripción, Importancia, Prioridad, Botón de fecha y hora, Fecha, y Hora.

La clase Bank: maneja un ArrayList de la clase tarea, que se inicializa en su constructor, y en ese ArrayList, se tiene toda la información de las tareas del usuario. Además, también tiene todos los métodos para filtrar y ordenar a conveniencia el arreglo de tareas.

2DoList – Manual de usuario

Entorno de entrada.

Al entrar a la aplicación se encontrará con una de las siguientes vistas, esto depende de la posición en la que se encuentre el teléfono.

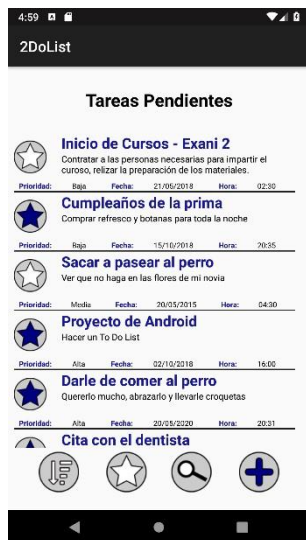


Figura 2. Vista vertical.

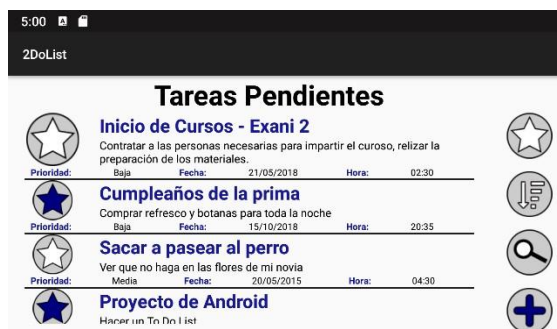


Figura 3. Vista Horizontal.

Agregar una nueva tarea


Para agregar una nueva tarea es necesario pulsar el botón de agregar , este nos mandará a una nueva ventana en donde podremos llenar los parámetros necesarios de la tarea a agregar.



Figura 4. Agregar tarea

Es necesario que se llenen los parámetros que se requieran recordar de la tarea, dando click en “Insertar título” y “Describir tarea”, podremos añadir título y descripción respectivamente.

- Si nuestra tarea es importante basta con pulsar el símbolo de ⭐ para que la aplicación la interprete la importancia, una vez que la app adquiera la importancia la estrella se tornará Azul 🌟.
- Para seleccionar la prioridad de la tarea, es necesario picar en la pestaña de prioridad y se nos desplegará una lista con las opciones disponibles (Baja, Media o Alta), solo seleccionamos la de nuestro interés y la prioridad será fijada.



Figura 5. Selección de prioridad.

- Para poder agregar una fecha de vencimiento es necesario activar el switch y los botones de fecha y hora se activarán.



Figura 6. Selección de

vencimiento.

Para añadir la hora y fecha es necesario pulsar el botón de cada uno de ellos y seleccionar los datos deseados.

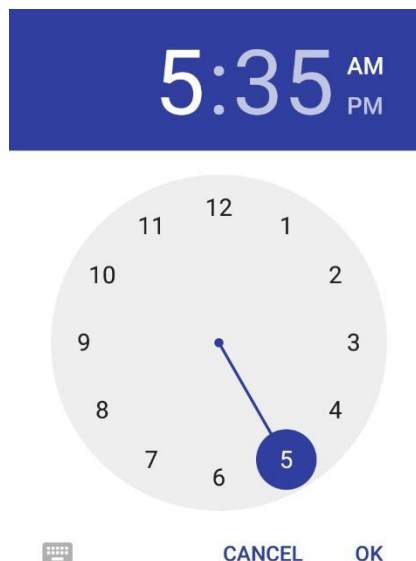


Figura 7. Selección de hora



Figura 8. Selección de fecha

Para finalizar y añadir la tarea es necesario clickear en +.

Filtrado por importancia

Para activar o desactiva un filtro en 2DoList es necesario clickear sobre su icono, en este caso el filtro de importancia solo muestra las tareas que son importantes, es decir las que fueron marcadas con ★. Una vez que el filtro fue activado, la app mostrará un mensaje de que el filtro fue aplicado y la lista principal será actualizada con las tareas que cumplen dichas características.



Figura 9. Filtro de importancia activado.

Filtro por Orden

Para hacer el uso de este filtro es necesario configurarlo, para lograr hacer eso es necesario dejar apretado el botón de ⓘ, con esta acción se nos desplegará una nueva ventana de configuración.

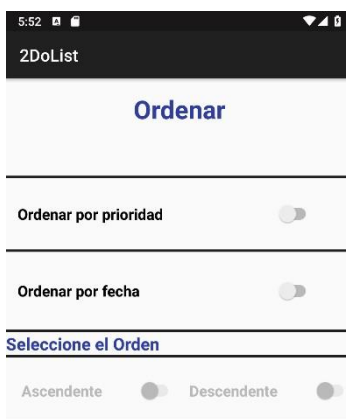



Figura 10. Ordenar – Configuración

•

Al entrar a la ventana de ordenar es necesario seleccionar las características de nuestro orden deseado, podemos ordenar por fecha o por prioridad, una vez seleccionada nuestra característica, se procede a seleccionar el orden (Ascendente o Descendente). • Para aplicar el filtro solo basta con presionar el botón de back.

- El filtro se puede activar o desactivar clickeando sobre , la configuración seleccionada se guardará hasta que entremos de nuevo a modificar el filtro.

Filtrado por búsqueda y coincidencias



Para hacer el uso de este filtro es necesario configurarlo, para lograr hacer eso es necesario dejar apretado el botón de , con esta acción se nos desplegará una nueva ventana de configuración.



Figura 11. Filtrado – Coincidencias

- Para filtrar cada uno de los parámetros es necesario activarlo, esto lo hacemos clickeando sobre el switch, es importante mencionar que todos los filtros que estén activados se aplicarán de manera simultánea.
- Solo activa las características que desees encontrar en tus tareas e ingresa los parámetros necesarios para que el filtro haga la búsqueda.
- Para aplicar el filtro solo basta con presionar el botón de back.
- El filtro se puede activar o desactivar clickeando sobre , la configuración seleccionada se guardará hasta que entremos de nuevo a modificar el filtro.


Aplicación de múltiples filtros

En 2DoList se pueden aplicar todos los filtros de manera simultánea, solo basta con configurar los filtros y activar los que desees, por ejemplo, podemos tener activos los siguientes:



Figura 12. Todos los filtros activos.

Edición de una tarea ya creada

Para modificar una tarea ya creada basta con pulsar sobre ella, en este se abrirá una pestaña similar a la de añadir tarea, solo tienes que modificar el parámetro que cambió y pulsar  para guardar los cambios.

Eliminación de una tarea

Para eliminar una tarea basta con dejar presionado sobre ella, esta acción desplegará una ventana emergente que nos preguntará si deseamos eliminar la tarea, si nuestras repuestas es “Aceptar”, la tarea se eliminará de manera instantánea.

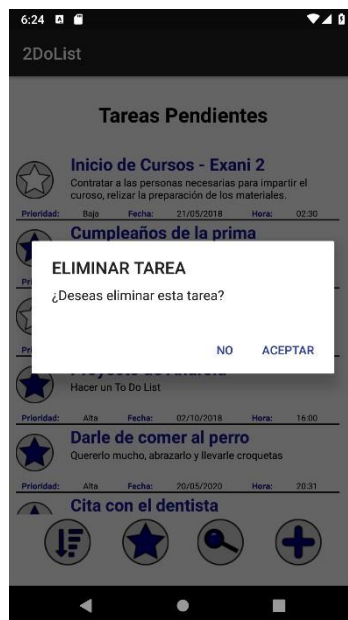


Figura 13. Eliminar tarea.

Conclusión grupal

La elaboración de esta app nos ayudó a fortalecer nuestros conocimientos de programación ya que fuimos viendo como estructurar de manera lógica todo el código, la coordinación como equipo fue algo primordial y algo que supimos controlar desde el primer día.

La aplicación resultó como esperábamos, todas nuestras ideas fueron plasmadas de manera correcta y lógica, los errores que se nos fueron presentando fueron solucionados de manera rápida ya que manejamos todas nuestras operaciones con funciones, esto nos facilitó mucho el análisis del código.

La manera en la que fue plasmado nuestro código resultó ser muy diferente a la de nuestros compañeros, ya que nosotros omitimos el ID, ya que lo único que hacemos es adquirir la posición actual de la tarea, y para relacionar la misma tarea de distintas listas (original y filtrado), se utilizó una función para encontrar dicha tarea mediante comparaciones.

En vez de utilizar 5 actividades utilizamos 4, ya que la ventana de agregar y modificar tarea es la misma, esto se consiguió con las respuestas de `OnActivityResult`.

Como extras nuestra app tiene diseño propio, ya que nos sobró tiempo para diseñar nuestros propios botones, logotipo y la estructura de nuestro `RecyclerView`, de igual forma implementamos la `Fragments` para la adquisición de la fecha y hora.

Conclusiones individuales

Dzay Villanueva Luis Fernando

El trabajo en equipo fue primordial, llegué a la conclusión de que para trabajos de equipo en programación, ponerse de acuerdo con los nombres de todas las variables es primordial, así como crear funciones para que sea más entendible el código al momento de juntar partes.

Para el caso de los filtros, se optimizó mucho el código respecto a los demás equipos, porque para hacer un AND entre los mismos, lo que se hizo fue manejar dos bancos, un banco principal y otro banco auxiliar que manejara todos los filtros. Por lo que si por ejemplo se quería tener filtro por título e importancia, se extraía el arreglo del filtro por título, y a ese arreglo, se le aplica el filtro de importancia, obteniendo así el AND entre los dos filtros.

De igual manera, al ordenar las tareas por fecha, para reducir el código lo que se hizo es manejar toda la fecha incluyendo: Día, mes y año; en un solo número, para así poder comparar simplemente si una fecha es mayor o menor.

González Ávila Víctor Antonio

La elaboración de 2DoList resultó en un gran aprendizaje para mí, nunca había interactuado en un Java; mediante lectura y análisis de videos fui aprendiendo la sintaxis y análisis del código, sin embargo, este proyecto me ayudó a fortalecer el conocimiento que ya tenía de POO.

La aplicación resulto mejor de lo que esperábamos, esto se debe a que empezamos a actuar desde el primer día en que el proyecto fue marcado, nos metimos de lleno a la bibliografía y a la investigación; la primera semana estábamos atrasados con respecto a otros equipos, sin embargo, ese fin de semana remontamos y adquirimos un nivel igual o mayor que al que tenían otros. Prácticamente nuestra primera semana fue de experimentación.

La mayoría de la aplicación la hicimos de manera simultánea, esto fue por nuestro proceso de aprendizaje, de ahí nos dividimos al llegar a los filtros, donde aporté el filtro de Ordenar y el de importancia. Uno de los mayores problemas que tuvimos fue adquirir la fecha como entero, esto con la finalidad de adquirir una jerarquía (Orden). Esta parte la solucionó mi compañero.

Otro aporte importante fue la implementación del RecyclerView, el diseño de los botones y la creación del logo.