

**INSTITUTO POLITÉCNICO NACIONAL**

**ESCUELA SUPERIOR DE CÓMPUTO**

**ALUMNO:**

**ALDAVERA GALLAGA IVÁN**

**APPLICATION DEVELOPMENT FOR MOBILE DEVICES**

**PROFESOR:**

**ALEJANDRO SIGFRIDO CIFUENTES ÁLVAREZ**

**TÍTULO DEL REPORTE**

**NOTIFICACIONES**

**FECHA DE REALIZACIÓN:**

**03 – JUNIO – 2020**

**VERSIÓN DEL REPORTE:**

**V1**

**ÍNDICE**

OBJETIVO……………………………………………………………………………..1

CONCEPTOS………………………………………………………………………….1

DESARROLLO…………………………………………………………………………2

Ejemplo 1……………………………………………………………………….2

Ejercicio 1……………………………………………………………………….2

Ejemplo 2……………………………………………………………………….2

Listado de Software……………………………………………………………3

Listado de Hardware…………………………………………………………..3

PRUEBAS………………………………………………………………………………3

Ejemplo 1……………………………………………………………………….3

Ejercicio 1……………………………………………………………………….4

Ejemplo 2……………………………………………………………………….4

CONCLUSIONES……………………………………………………………………..5

BIBLIOGRAFÍA………………………………………………………………………..5

**OBJETIVO**

Entender qué con las notificaciones, su importancia en una aplicación Android y cómo implementarlas en una.

**CONCEPTOS**

Las notificaciones son avisos de eventos importantes que ocurren en segundo plano. Las notificaciones se visualizan en el área de notificaciones en forma de icono descriptivo. Para ver la notificación se arrastra la barra de notificaciones hacia abajo y enseguida se muestran las notificaciones acumuladas hasta el momento.

Las notificaciones tienen cuatro partes fundamentales: El título, un contenido, el ícono y el tiempo de ocurrencia. Se pueden añadir más elementos, por ejemplo iconos combinados, la cantidad de notificaciones, botones, y otros.

Las notificaciones de aviso aparecen con prioridad alta o porque se definió que la notificación se proyecte de esta forma. Las llamadas entrantes o alarmas son eventos que producen notificaciones de aviso.

Las notificaciones utilizan la clase Notification, pero para tener compatibilidad con versiones anteriores se usa NotificationCompat. La clase PendingIntent contiene los datos de un Intent y la actividad que iniciará al presionarse la notificación. Las notificaciones pueden eliminarse desde la interfaz arrastrándola hacia el drawer. También, se puede digitar el botón para la eliminación de todas las notificaciones. Las notificaciones pueden expandir su contenido si se aplica el estilo BigTextStyle, que permite ampliar el detalle del texto del contenido. Las notificaciones permiten visualizar progresos de operaciones en curso relacionadas a otras aplicaciones. Para crear una notificación de aviso se utiliza el método setFullScreenIntent().

Las notificaciones se muestran en la pantalla de bloqueo si el usuario así lo decide en la configuración de las preferencias de la plataforma. Las siguientes constantes definen si las notificaciones se muestran en la pantalla de bloqueo:

VISIBILITY\_PUBLIC Muestra todo el contenido de la notificación en la pantalla de bloqueo.

VISIBILITY\_PRIVATE Muestra el icono, el título y detalles mínimos de la notificación.

VISIBILITY\_SECRET No muestra ninguna información de la notificación.

La jerarquía de la clase Notification: public class Notification extends Object implements Parcelable java.lang.Object -> android.app.Notification .

Esta clase representa la forma en que una notificación persistente se presenta al usuario utilizando el NotificationManager. Además, se ha agregado la clase Notification.Builder para facilitar la construcción de la notificación.

Métodos públicos:

|  |  |  |
| --- | --- | --- |
| Notification | clone() | Crea y regresa una copia de este objeto |
| int | describeContents() | Describe de objetos contenidas en la representación referenciada de la instancia Parcelable |
| String | getGroup() | Obtiene la clave para agrupar esta notificación en un grupo o pila con otras notificaciones en los dispositivos que soporten tal prestación |
| Icon | getLargeIcon() | El icono grande que se muestra en la vista del contenido de la presente notificación |
| Icon | getSmallIcon() | El icono pequeño que representa esta notificación en la barra de estado y el contenido. |
| String | getSortKey() | Obtiene la clave de clasificación que ordena la presente notificación entre otras notificaciones del mismo paquete. |
| String | toString() | Regresa la cadena que representa a este objeto. |
| void | writeToParcel(Parcel p, int flags) | Empaqueta |

**DESARROLLO**

Ejemplo 1

Para el desarrollo del ejemplo se debe desarrollar una aplicación en la que cada vez que se presione un botón salga un mensaje de alerta y se lleve a cabo el incremento en un contador que parte de 0.

También se despliega una notificación que al presionarse vuelve a cero el contador indicando que se atendió la notificación pero sin quitar el mensaje.

Ejercicio 1

Similar al anterior, pero debe verse en 0 el contador si la lista de notificaciones se encuentra vacía.

Ejemplo 2

Este ejemplo es similar a los anteriores, solo que sin contador y con un mensaje en un TextView y borrándose las notificaciones una vez regresando atrás.

**LISTADO DE SOFTWARE:**

* Java Development Kit 8: Necesario para el desarrollo de aplicaciones Android.
* Android Studio 4.0: IDE para el desarrollo de aplicaciones Android
* Windows 10: Sistema operativo sobre el que corren los anteriores programas.
* Android 10. Sistema Operativo sobre el que corre la aplicación

**LISTADO DE HARDWARE:**

* Laptop HP con GB RAM DDR4 y procesador Intel Core i3
* Dispositivo Móvil Motorola One con 4GB de RAM y procesador SoC Qualcomm Snapdragon 625
* Cable tipo C para la transferencia de datos.

**PRUEBAS**

Ejemplo 1:

Tuve algunos problemas con la ejecución del primer ejemplo, pues al parecer muchas de las bibliotecas ya no se encuentran en el mismo lugar y por lo tanto las funciones del código aparecen como erróneas, sin embargo, pude arreglarlo sustituyendo algunas líneas de código como las siguientes:

import androidx.core.app.NotificationCompat;

sustituida por:

import android.support.v4.app.NotificationCompat;

La siguiente línea también me arrojaba un error:

getMenuInflater().inflate(R.menu.activity\_main, menu);

Para lo cual tuve que crear una carpeta llamada ‘menu’ dentro de la carpeta res/values.

Al correr la aplicación en mi dispositivo móvil obtuve lo siguiente:



Aún así, al presionar el botón no funcionaba nada.

Ejercicio 1:

Dados los problemas con el Ejemplo 1, no me fue posible realizar este ejercicio.

Ejemplo 2:

Para la codificación del Ejemplo 2 todo fue muy bien, ningún error, solo que al momento de correr la aplicación en mi dispositivo móvil tampoco funcionaba nada correctamente.



E incluso la aplicación se detuvo después de unos segundos.

**CONCLUSIONES**

Tuve muchos problemas con la correcta codificación y ejecución de las notificaciones y de las aplicaciones mismas en esta práctica, por lo que indagué, se debe a que en las versiones recientes de Android hubo algunos cambios en el desarrollo de notificaciones, razón por la cual ahora es un poco diferente la utilización de las funciones de las bibliotecas que se encargan de gestionar las notificaciones.

**BIBLIOGRAFÍA**

* Archivo Android Notificaciones, Cifuentes, A.. ESCOM-IPN