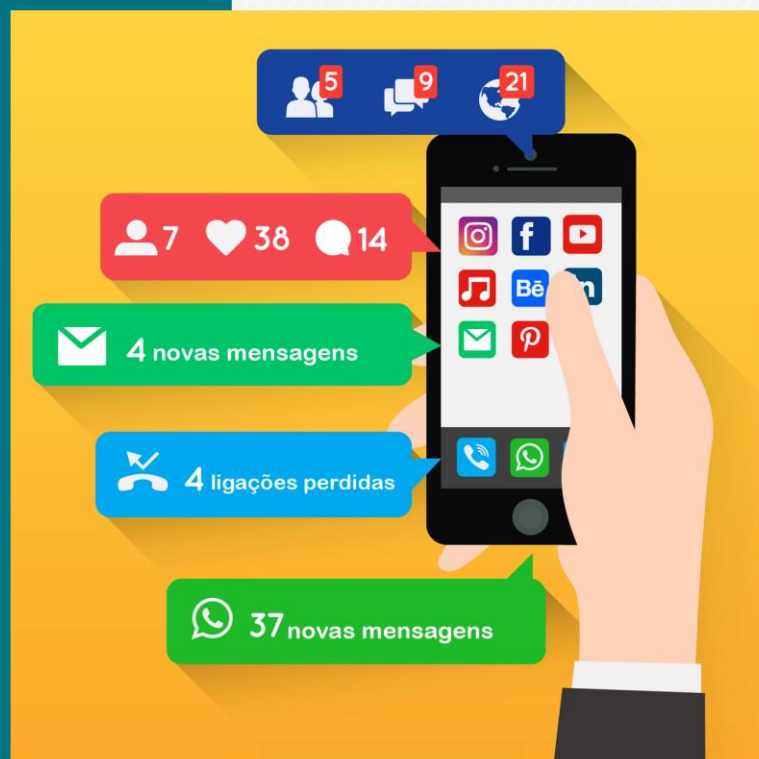


AGENDA 7

NOTIFICAÇÕES



GEEaD - Grupo de Estudos de Educação a Distância
Centro de Educação Tecnológica Paula Souza

GOVERNO DO ESTADO DE SÃO PAULO
EIXO TECNOLÓGICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO TÉCNICO EM DESENVOLVIMENTO DE SISTEMAS
PROGRAMAÇÃO MOBILE I

Expediente

Autor:

GUILHERME HENRIQUE GIROLI

Revisão Técnica:

Eliana Cristina Nogueira Barion

Revisão Gramatical:

Juçara Maria Montenegro Simonsen Santos

Editoração e Diagramação:

Flávio Biazim

São Paulo – SP, 2020



Uma notificação é uma mensagem criada por um aplicativo, em primeiro plano ou em execução em segundo plano, e serve para chamar a atenção do usuário do sistema operacional Android para uma ocorrência que foi gerada no aplicativo ou fora dele.

O sistema Android é responsável por gerar a notificação na área de notificação, que fica próximo ao relógio do dispositivo. E quando o usuário deseja visualizar maiores informações sobre a notificação, é necessário abrir a gaveta de notificações, onde ela é exibida de maneira completa.

A figura 4 mostra a área de notificação e a figura 5 mostra a gaveta de notificações.

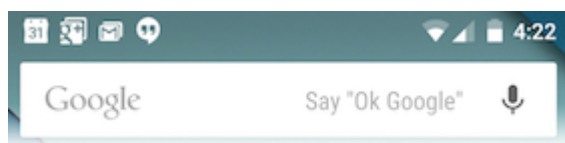


Figura 4 – Área de notificação. Disponível em:
<https://developer.android.com/guide/topics/ui/notifiers/notifications?hl=pt-br>

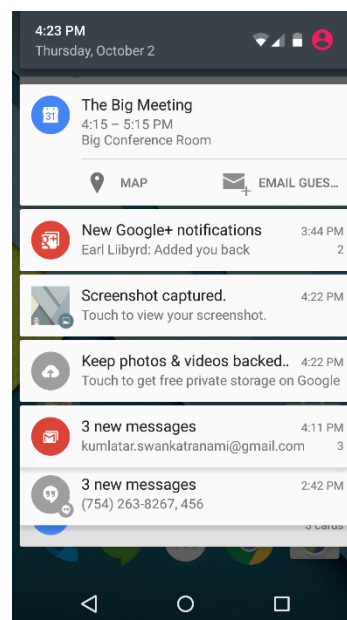


Figura 5 – Gaveta de notificações.
Disponível em:
<https://developer.android.com/guide/topics/ui/notifiers/notifications?hl=pt-br>

Para criar uma notificação no sistema operacional Android é necessário trabalhar com algumas regras. Desta forma o Android Studio oferece pacotes de classes que auxiliam o desenvolvimento das notificações.

A classe `NotificationCompat()` foi desenvolvida para facilitar a criação das notificações. Ela permite o desenvolvimento de notificações expandidas e notificações simples de acordo com a necessidade do desenvolvedor e da versão do sistema operacional Android.

Ela oferece suporte para que o desenvolvedor construa os recursos mais simples de uma notificação, como o título, ícones e a mensagem. E oferece suporte aos recursos mais avançados como notificações expandidas, que possuem botões para inúmeras utilizações. A figura 6 mostra um exemplo de notificação expandida.

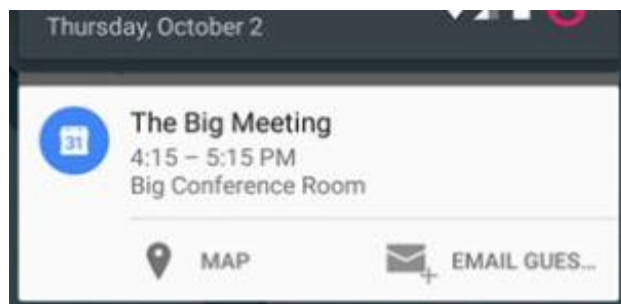


Figura 6 – Notificação expandida, com dois botões.

Vale ressaltar que as notificações que possuem botões, somente são exibidas em sua forma completa, ou seja, mostrando a mensagem e os botões nas versões superiores ao Android 4.1.

No processo de construção da notificação, utilizamos alguns métodos para customizar a sua exibição para o usuário. A figura 7 apresenta uma pequena lista dos principais métodos, e na documentação oficial do Android Studio, disponível no link:

<https://developer.android.com/reference/android/support/v4/app/NotificationCompat.Builder>.

Podemos encontrar a lista completa dos métodos presentes na classe `NotificationCompat()`.

`setContentText(CharSequence text)`

Definir o texto (segunda linha) da notificação, em uma notificação padrão.

`setContentTitle(CharSequence title)`

Defina o título (primeira linha) da notificação, em uma notificação padrão.

`setSmallIcon(int icon, int level)`

Uma variante `setSmallIcon(int)` disso requer um parâmetro de nível adicional para quando o ícone é um `LevelListDrawable`.

`setSmallIcon(int icon)`

Defina o pequeno ícone para usar nos layouts de notificação.

`setPriority(int pri)`

Defina a prioridade relativa para essa notificação.

Figura 7 – Métodos da classe `NotificationCompat()`.

Com os métodos apresentados anteriormente, já conseguimos gerenciar a criação de uma notificação básica, para ser exibida na área e na gaveta de notificações. O que falta é aplicar os processos que serão executados quando o usuário clicar na notificação gerada.

Podemos definir que uma determinada notificação seja persistente, mesmo que o usuário a visualize e clique sobre ela, ela continue na área de notificação. Ou podemos fazer com que ela tenha um ciclo de vida, ou seja, será eliminada da área e gaveta de notificações, após o usuário clicar sobre ela. Este recurso é definido através do método `setAutoCancel()`, como na figura 8.

```
setAutoCancel(boolean autoCancel)
```

Definir esse sinalizador fará com que a notificação seja cancelada automaticamente quando o usuário clicar no painel.

Figura 8 – Método `setAutoCancel()` da classe `NotificationCompat()`.

Para finalizar a etapa de criação da notificação vamos informar para o Android qual Activity será aberta quando o usuário clicar na notificação na gaveta de notificações.

O método `setContentIntent()` é o responsável por atribuir essa função a notificação. Esse método trabalha com o recebimento de um `PendingIntent`.

O `PendingIntent` é criado para que o gerenciador do Android consiga executar uma determinada Activity do seu projeto, chamando ela na tela do dispositivo quando o usuário clicar na notificação. A figura 9 apresenta o `setContentIntent()`.

```
setContentIntent(PendingIntent intent)
```

Fornecer um `PendingIntent` para enviar quando a notificação for clicada.

Figura 9 – Método `setContentIntent()` da classe `NotificationCompat()`.

Após a criação da notificação através da classe `NotificationCompat()`, o desenvolvedor atribui a função de exibição ao sistema operacional Android através do `NotificationManager()`.

A classe `NotificationManager()` trabalha em conjunto com o sistema operacional para executar a notificação criada anteriormente. As notificações podem conter mensagens, alertas luminosos (caso o dispositivo tenha LEDs para notificações) e até mesmo alertas sonoros e vibrações.

Utilizamos o método *Notify* do objeto criado através do `NotificationManager()` para inserir nossa notificação na área de notificação do dispositivo. Cada notificação necessita de um identificador, ou seja, um número único que diferencia uma notificação de outra. No caso de uma notificação ser gerado com o mesmo número de uma que já esteja na área de notificação, a mais antiga é sobrescrita pela mais nova.

Aplicando a notificação no projeto

Vamos desenvolver um projeto para testar as notificações, é importante ressaltar que vamos utilizar uma API 24, para atender os novos padrões de notificações. Para essa etapa é fundamental instalar e criar uma AVD na API 24. O projeto deve conter as seguintes características:

Application Name: Notificacoes

API: 24

Activity inicial: Empty Activity

Activity Name: Notificacao

Observação: Desmarque a opção “**Backwards compatibility (AppCompat)**”

Após desenvolver o projeto, crie mais um Activity com as seguintes características:

Activity: Empty Activity

Activity Name: Promocao

Observação: Desmarque a opção “**Backwards compatibility (AppCompat)**”

A figura 10 mostra a estrutura do nosso projeto.

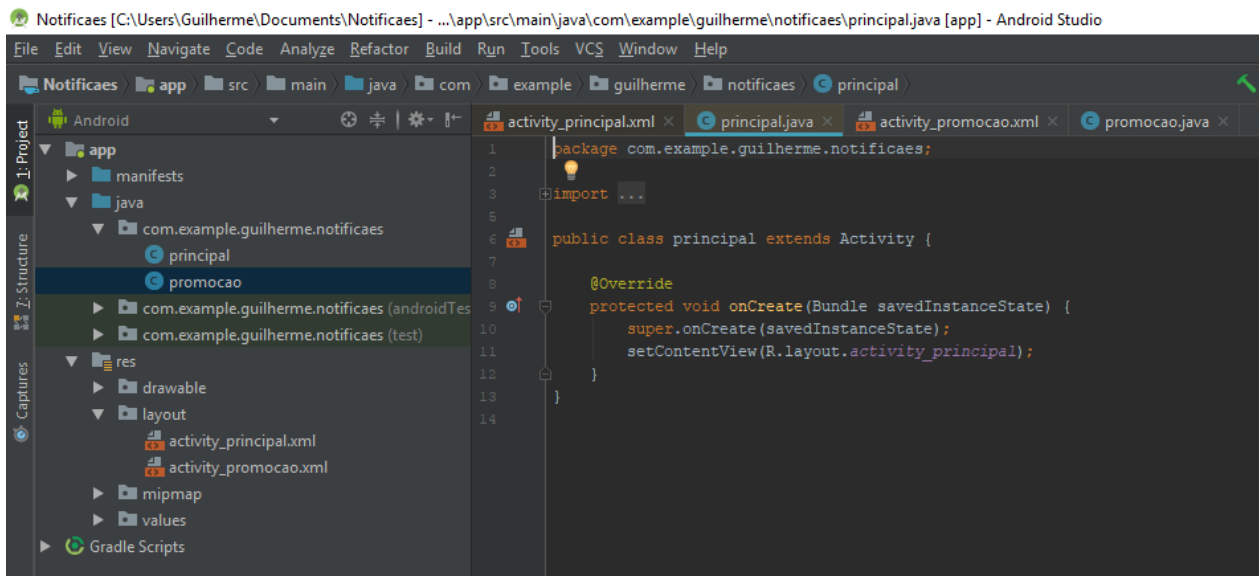


Figura 10 – Estrutura do projeto “Notificações”.

Na Activity “principal” vamos desenvolver o seguinte Layout, como mostra a figura 11:

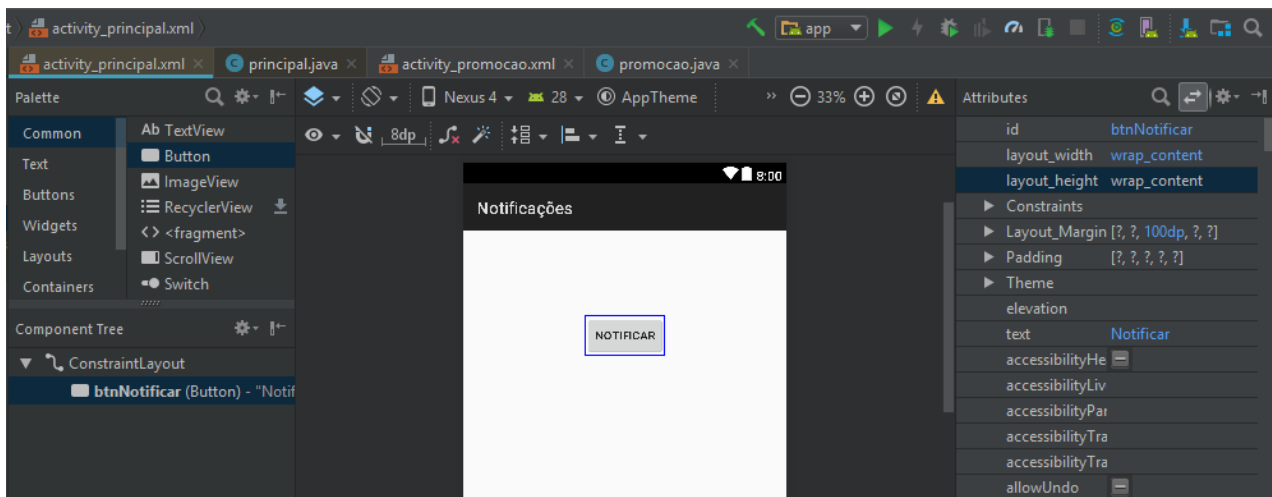


Figura 11 – Estrutura da Activity “principal”.

Na Activity “promocao” vamos desenvolver o seguinte Layout, como mostra a figura 12:

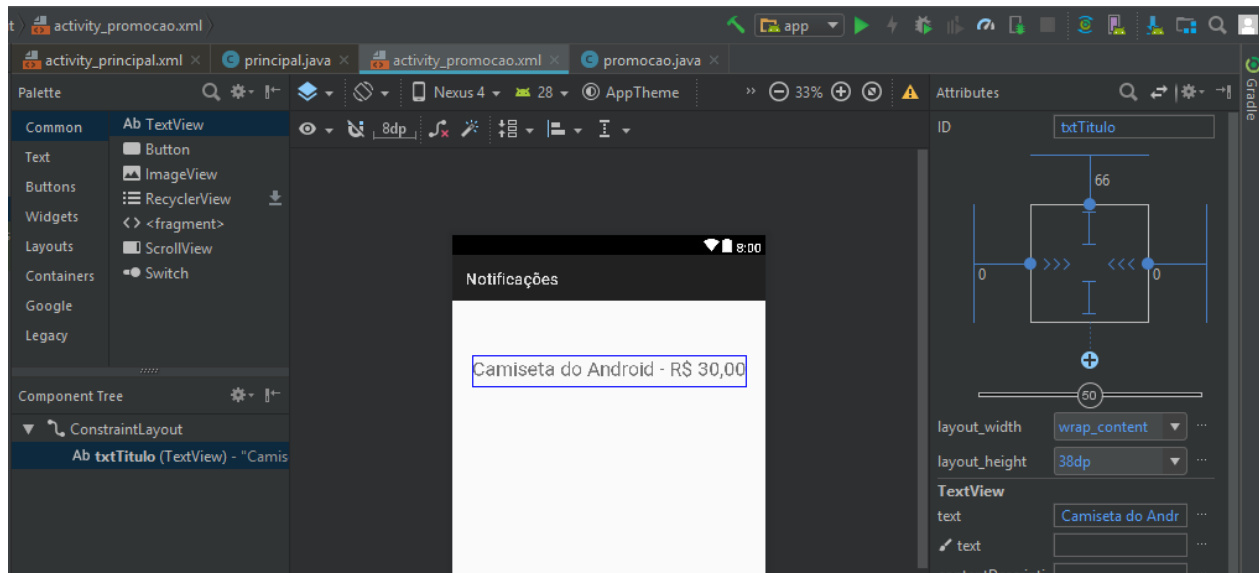
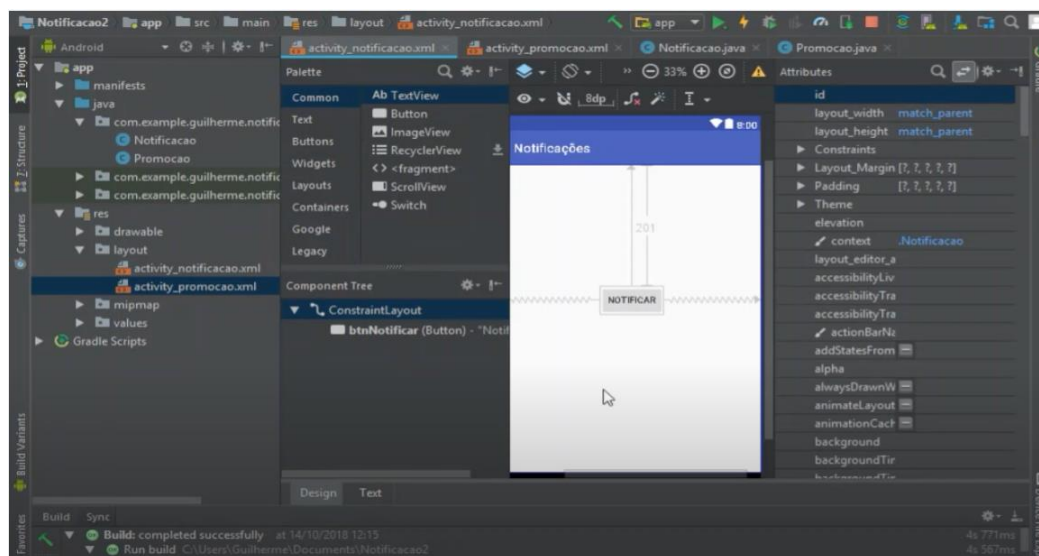


Figura 12 – Estrutura da Activity “promocao”.

O vídeo a seguir mostra o código da classe “Notificacao” e faz uma explicação sobre o processo de criar e gerar a notificação no dispositivo mobile do sistema operacional Android. O código utilizado no vídeo está disponível na sequência.




```

package com.example.guilherme.notificacoes;

import android.app.Notification;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.v4.app.NotificationCompat;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;

public class Notificacao extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_notificacao);

        Button btnNotificarProg = (Button) findViewById(R.id.btnNotificar);

        btnNotificarProg.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                String mensagem = "50% de desconto.";
                String titulo = "Promoção!";
                notificarUsuario(mensagem, titulo);

            }

            //método notificarUsuario
            //Parâmetros do tipo texto: Mensagem e Título
            private void notificarUsuario(String mensagem, String titulo){
                //Objeto "notificacao" para receber os parâmetros e construir
                //a estrutura da notificação
                NotificationCompat.Builder notificacao = new
                NotificationCompat.Builder(getApplicationContext());
                //Construindo o título da notificação
                notificacao.setContentTitle(titulo);
                //Construindo a mensagem da notificação
                notificacao.setContentText(mensagem);
                //Construindo a prioridade da notificação
                notificacao.setPriority(Notification.PRIORITY_HIGH);
                //Construindo o ícone da notificação
                notificacao.setSmallIcon(R.drawable.ic_launcher_foreground);
                //Quando o usuário clicar na notificação, ela será excluída da
                gaveta de notificações
                notificacao.setAutoCancel(true);
                //Construindo a intenção que chama a Activity promoção
                //quando o usuário clicar na mensagem
                Intent it = new Intent(getApplicationContext(), Promocao.class);
                //Utilizando o PendingIntent para que o Android consiga
                executar a Activity
                //do seu projeto
                PendingIntent pit =
                PendingIntent.getActivity(getApplicationContext(), 100,

```

```

        it, PendingIntent.FLAG_UPDATE_CURRENT);
        //Atribuindo na notificação a intenção que será executada,
quando o usuário
        //clicar na notificação
        notificacao.setContentIntent(pit);
        //Criando o objeto para gerenciar e gerar a notificação no
Android
        NotificationManager notificar = (NotificationManager)
getBaseContext().getSystemService(Context.NOTIFICATION_SERVICE);
        //Executando a notificação com o id fixo: 9000
        notificar.notify(9000, notificacao.build());
    }
}
}
}

```

Trabalhando com imagens no projeto

No desenvolvimento Android nos deparamos com algumas necessidades, entre elas a inserção de imagens. Para trabalhar com imagens utilizamos a ferramenta “ImageView”.

O ImageView trabalha com imagens, e vamos inserir a imagem utilizada nesta ferramenta na pasta “Drawable” do nosso projeto. No projeto de Bruna, é necessário a inserção de uma imagem referente a promoção de uma camiseta do Android. Vamos acompanhar o desenvolvimento desta etapa do projeto no vídeo a seguir.

