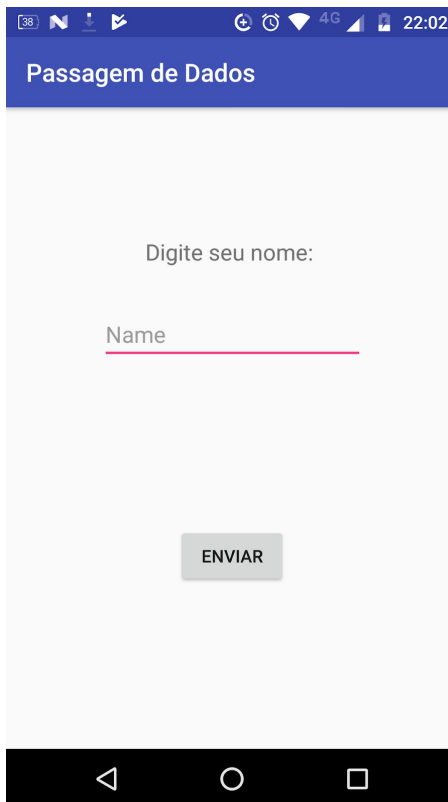


## **Tutorial Passagem de Dados entre Activities**

Hoje, veremos como passar dados entre diferentes Activities. Até agora, nossos apps eram compostos de apenas uma Activity, que respondia aos eventos modificando seus elementos. É uma abordagem válida, mas na maioria das vezes, um app terá mais de uma Activity, separando suas funções e muitas vezes elas deverão trocar dados. Vimos que devemos usar um Intent para isso. Vamos primeiro implementar um exemplo de como passar dados de um EditText para uma segunda Activity.

Comece clonando o seu repositório e configurando os dados do usuário. No seu repositório, crie uma pasta para o novo projeto com o nome **Passagem de dados**. Após isso crie um novo projeto no Android Studio e não se esqueça de mudar o local do projeto para a pasta que você criou.

Agora, modifique o layout da MainActivity para que fique parecida com a abaixo:



The screenshot shows a mobile application interface. At the top, there is a blue header bar with the title "Passagem de Dados". Below the header, the text "Digite seu nome:" is displayed. Underneath this text is a text input field with the placeholder "Name". At the bottom of the screen, there is a button labeled "ENVIAR". The interface is set against a light gray background. The status bar at the top of the screen shows various icons including signal strength, Wi-Fi, battery, and the time 22:02. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps buttons.

Vamos agora programar o comportamento da Activity:

```

1 package com.example.wendell.passagemdedados;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.util.Log;
7 import android.view.View;
8 import android.widget.EditText;
9 import android.widget.Toast;
10
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17     }
18
19     public void mandaMensagem(View v) {
20         EditText nome = findViewById(R.id.txtNome);
21         String oNome = nome.getText().toString();
22         if (oNome.matches("")) {
23             Toast toast = Toast.makeText(getApplicationContext(),
24                 "É preciso digitar um nome...", Toast.LENGTH_SHORT);
25             toast.show();
26             return;
27         }
28         Intent i = new Intent(MainActivity.this, Resultado.class);
29
30         Bundle bundle = new Bundle();
31         bundle.putString("nome", oNome);
32         i.putExtras(bundle);
33         startActivity(i);
34     }
35 }

```

Lembre-se de substituir o id do TextView de acordo com o id que você usou em sua interface. Repare que temos um novo elemento no código, o **Toast**. Um toast é uma pequena notificação que é mostrada para alertar o usuário, neste caso, que ele precisa digitar algo. Veja que criamos um Intent, onde passamos o contexto da aplicação e a atividade que será iniciada. Em seguida, criamos um Bundle para empacotar os dados que serão passados, neste caso, uma única String.

Agora, crie uma nova Activity vazia, que será onde receberemos os dados que passamos. Chame-a de **Resultado**. Para isso, clique com o botão direito no lado esquerdo do editor, sobre a estrutura do projeto e escolha *New/ Activity/ Empty Activity*. Modifique o layout para que fique parecido com a figura abaixo:



O código da Activity Resultado ficará assim:

```
1 package com.example.wendell.passagemdedados;
2
3 import android.content.Intent;
4 import android.support.v7.app.AppCompatActivity;
5 import android.os.Bundle;
6 import android.widget.TextView;
7
8 public class Resultado extends AppCompatActivity {
9
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
13         setContentView(R.layout.activity_resultado);
14
15         Intent intencao = getIntent();
16         Bundle pacote = intencao.getExtras();
17
18         String nome = pacote.getString("nome");
19         TextView txtResultado = (TextView) findViewById(R.id.txtResultado);
20         txtResultado.setText(String.format("Olá, %s!", nome));
21     }
22 }
```

Veja que no código, são criados um Intent e um Bundle, que através dos métodos `getIntent()` e `getExtras()`, são carregados com os dados enviados pela primeira Activity. Execute o app e verifique o seu funcionamento. Não esqueça de fazer a sincronização com o seu repositório remoto, através da trinca de comandos *git add - git commit - git push*.

Agora, vamos ver como funciona uma Intent implícita. Crie um novo projeto em uma nova pasta no seu repositório local. Dê o nome **Intent Implícita**. Adicione uma Activity vazia e modifique o layout para que fique como abaixo:



Depois, digite o seguinte código na MainActivity.java:

```

1 package com.example.wendell.intentimplcita;
2
3 import android.content.Intent;
4 import android.net.Uri;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.view.View;
8 import android.widget.EditText;
9 import android.widget.Toast;
10
11 public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main);
17     }
18
19     public void abrirUrl(View v) {
20         EditText url = (EditText) findViewById(R.id.txtUrl);
21         String aUrl = url.getText().toString();
22         if (aUrl.matches("")) {
23             Toast toast = Toast.makeText(getApplicationContext(),
24                 "Digite um endereço web...", Toast.LENGTH_SHORT);
25             toast.show();
26             return;
27         }
28         Intent intencao = new Intent(Intent.ACTION_VIEW, Uri.parse(aUrl.toLowerCase()));
29         startActivity(intencao);
30     }
31 }

```

Execute o aplicativo e tente abrir um endereço da web, por exemplo: <http://www.google.com.br>. Agora, tente abrir com o endereço sem http://. Vai gerar um erro. Ou seja, é obrigatório colocar o http://, o que nos obriga a validar a url digitada antes de criar o Intent e iniciar a atividade. A validação da url fica como exercício.

Voltaremos a usar a passagem de dados futuramente. Agora, vamos estudar a persistência de dados simples, usando o SharedPreferences.

Crie um novo projeto, novamente em uma nova pasta em seu repositório e chame-o de **SharedPreferences**. Modifique o layout da MainActivity como abaixo:



Agora, digite o código:

```

1 package com.example.wendell.sharedpreference;
2
3 import android.content.Context;
4 import android.content.SharedPreferences;
5 import android.support.v7.app.AppCompatActivity;
6 import android.os.Bundle;
7 import android.util.Log;
8 import android.view.View;
9 import android.widget.EditText;
10 import android.widget.Toast;
11
12 import static android.app.PendingIntent.getActivity;
13
14 public class MainActivity extends AppCompatActivity {
15
16     EditText valor;
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
21         valor = (EditText) findViewById(R.id.txtValor);
22     }
23
24     public void salvar(View v) {
25         SharedPreferences arquivo = getPreferences(Context.MODE_PRIVATE);
26         String oValor = valor.getText().toString();
27         if(oValor.matches("")) {
28             Toast toast = Toast.makeText(MainActivity.this,
29                 "Digite algo...", Toast.LENGTH_SHORT);
30             toast.show();
31             return;
32         }
33         SharedPreferences.Editor editor = arquivo.edit();
34         editor.putString("valor", oValor);
35         editor.commit();
36     }
37
38     public void carregar(View v) {
39         SharedPreferences arquivo = getPreferences(Context.MODE_PRIVATE);
40         String oValor = arquivo.getString("valor", "Nada...");
41         valor.setText(oValor);
42         Log.i("oValor: ", oValor);
43     }
44 }

```

Associe os métodos aos botões correspondentes e rode o app. Verifique seu funcionamento, salvando um valor, fechando o app e carregando-o.

Agora é sua vez... Vamos aproveitar a ideia do jogo de adivinhação e modificá-lo para salvarmos o placar. Crie um novo projeto para o novo jogo.

- Modifique o jogo para que o número gerado seja entre 1 e 1000;
- A cada palpite do jogador, avise-o se o palpite foi maior ou menor que o número da sorte;
- O jogo prossegue até o jogador acertar o palpite;
- Conte o número de tentativas que o jogador levou para acertar o número;
- Ao final, salve o placar usando um SharedPreferences;
- Adicione ao app, além da Activity do jogo, uma outra Activity para verificar o placar;
- No placar, deve ser mostrado o melhor resultado já conseguido e um histórico com os resultados das 5 últimas partidas.