

```

package com.example.debugplayground

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.fillMaxSize
import androidx.compose.material3.MaterialTheme
import androidx.compose.material3.Surface
import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Modifier
import androidx.compose.ui.tooling.preview.Preview
import com.example.debugplayground.ui.theme.DebugPlaygroundTheme
import javafx.scene.paint.Material
import javafx.scene.text.FontWeight
import jdk.nashorn.internal.runtime.regexp.joni.Warnings
import java.lang.Exception
import java.lang.RuntimeException
import java.lang.reflect.Modifier
import javax.swing.GroupLayout.Alignment
const val TAG = "DebugPlayground"
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            DebugPlaygroundTheme {
                // A surface container using the 'background' color
                from the theme
                App() //Extrair a função utilizando as tecla ctrl+alt+M
            }
        }
    }
}

private fun App() { //mover a função da classe
    Surface(
        modifier = Modifier.fillMaxSize(),
        color = MaterialTheme.colorScheme.background
    ) {
        Column(
            //alinhamento vertical e horizontal
            verticalArrangement = Arrangement.SpaceEvenly,
            horizontalAlignment = Alignment.CenterHorizontally
        ) {
            Greeting("Playground") //Alterando a Preview
            //adicionar ação para o botão
            ActionButton(
                text = "Debug",
                buttonColors = DebugButtonColors(),
                modifier = Modifier.fillMaxWidth(0.5f)
            ) {
                Log.d(TAG, "Clicou em Debug")
            }
            ActionButton(
                text = "Info",
                buttonColors = InfoButtonColors(),
                modifier = Modifier.fillMaxWidth(0.5f)
            ) {
                Log.i(TAG, "Clicou em Info!")
            }
        }
    }
}

```

```

        ActionButton(
            text = "Warning",
            buttonColors = WarningButtonColors(),
            modifier = Modifier.fillMaxWidth(0.5f)
        ){
            Log.w(TAG, "Clicou em Warning!",)
        }
        ActionButton(
            text = "Error",
            buttonColors = ErrorButtonColors(),
            modifier = Modifier.fillMaxWidth(0.5f)
        ){//tratar ação para evitar de derrubar App
            try{
                throw RuntimeException("Clicou em ERROR")
            }catch (ex : Exception){
                Log.e(TAG, "${ex.message}",)
            }
        }
    }
}

//Cria a função ActionButton para trabalhar as ações dos botões com
três parâmetros diferentes
@Composable
fun ActionButton(
    text:String,
    buttonColors:ButtonColors =
ButtonDefaults.buttonColors(),//encapsulando as cores dos botões
    modifier: Modifier = Modifier, //modificando o botão estado vazio
    block: () -> Unit,//passando bloco como parametro para instanciar
o botão
){
    //informar o que tem o composable
    ElevatedButton(
        onClick=block,
        shape = RoundedCornerShape(5.dp),//botões com cantos
arredondados
        colors = buttonColors,
        modifier = modifier //modificando o botão
    ){
        Text(text = text)
    }
}

//Criando Preview do ActionButton
@Preview( showBackground = true , widthDp = 120)
@Composable
fun ActionButtonPreview(){
    ActionButton(
        text = "Action"){
    }
}

//Criar a função preview
@Preview(showBackground=true, widthDp=320, heightDp = 480)//alterando
o layout do App
@Composable
fun AppPreview(){

```

```
        DebugPlaygroundTheme{
            App()//chamar a função App
        }
    }

@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Welcome to $name!", //Concatenação de String
        //Customizar o Tema
        style = MaterialTheme.typography.bodyLarge.copy(
            fontWeight= FontWeight.Bold
        ),
        color = MaterialTheme.colorScheme.secondary
    )
}

@Preview(showBackground = true)
@Composable
fun GreetingPreview() {
    DebugPlaygroundTheme {
        Greeting("Android")//Alterando a Preview na função, para que a
        mesma possa funcionar
    }
}
```