

```

package com.exemplocrud;

import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.navigation.NavController;
import androidx.navigation.Navigation;
import androidx.navigation.ui.AppBarConfiguration;
import androidx.navigation.ui.NavigationUI;

import com.exemplocrud.databinding.ActivityMainBinding;
import com.google.android.material.bottomnavigation.BottomNavigationView;

public class CadastroAlunoActivity extends AppCompatActivity {

    private ActivityMainBinding binding;
    private BottomNavigationView navView;

    private EditText nome;
    private EditText cpf;
    private EditText telefone;

    private AlunoDao dao;

    //PARTE VI
    private Aluno aluno = null;

    private AppBarConfiguration appBarConfiguration; // Adicionando
    como variável de instância

    //ONCREATE
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        // No CadastroAlunoActivity setar o título
        getSupportActionBar().setTitle(R.string.title_home);

        // Vinculando os campos do layout com as variáveis do Java
        nome = findViewById(R.id.editNome);
        cpf = findViewById(R.id.editCPF);
        telefone = findViewById(R.id.editTelefone);

        dao = new AlunoDao(this);

        //PARTE VI ATUALIZAR
        Intent it = getIntent(); //pega intenção
        if(it.hasExtra("aluno")){
            aluno = (Aluno) it.getSerializableExtra("aluno");

```

```

        nome.setText(aluno.getNome().toString());
        cpf.setText(aluno.getCpf());
        telefone.setText(aluno.getTelefone());
    }

    navView = findViewById(R.id.nav_view);
    AppBarConfiguration appBarConfiguration = new
AppBarConfiguration.Builder(
        R.id.navigation_home, R.id.navigation_dashboard,
        R.id.navigation_notifications)
        .build();
    NavController navController =
Navigation.findNavController(this,
        R.id.nav_host_fragment_activity_main);
    NavigationUI.setupActionBarWithNavController(this,
        navController, appBarConfiguration);
    NavigationUI.setupWithNavController(binding.navView,
        navController);

    // Encontre o botão de salvar pelo ID
    Button salvarButton = findViewById(R.id.button);

    // Defina um ouvinte de clique para o botão de salvar
    salvarButton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            // Chame o método salvar quando o botão for clicado
            verificarENomeESalvar();
        }
    });

    navView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem
item) {
            int itemId = item.getItemId();

            if (itemId == R.id.navigation_home) {
                // Finalizar todas as atividades existentes
                //finish();
                Intent it = new Intent(CadastroAlunoActivity.this,
                CadastroAlunoActivity.class);
                startActivity(it);
                return true;
            } else if (itemId == R.id.navigation_dashboard) {
                // Lógica para a tela de listar alunos (Dashboard)
                //finish();
                startActivity(new
                Intent(CadastroAlunoActivity.this, ListarAlunos.class));
                return true;
            } else if (itemId == R.id.navigation_notifications) {
                // Finalizar todas as atividades existentes
                return true;
            }
            return false;
        }
    });
}

```

```

// Método para verificar o nome e salvar se estiver correto
private void verificarENomeESalvar() {
    String nomeAluno = nome.getText().toString();

    if (nomeAluno.isEmpty()) {
        Toast.makeText(this, "Por favor, insira um nome.",
Toast.LENGTH_SHORT).show();
    } else {
        // Verificação do nome
        if (nomeAluno.equalsIgnoreCase("NomeDesejado")) { //
Substitua "NomeDesejado" pelo nome desejado para verificar
            salvar();
        } else {
            Toast.makeText(this, "Nome incorreto!",
Toast.LENGTH_SHORT).show();
        }
    }
}

// Mantenha o método salvar original
private void salvar() {
    //Se o aluno não existe e não veio do atualizar, pega e
    cadastrar ele
    if(aluno==null) {
        aluno = new Aluno();
        aluno.setNome(nome.getText().toString());
        aluno.setCpf(cpf.getText().toString());
        aluno.setTelefone(telefone.getText().toString());
        long id = dao.inserir(aluno); // Inserir o aluno
        //MENSAGEM
        Toast.makeText(this, "Aluno inserido com id: " + id,
Toast.LENGTH_SHORT).show();
    }
    else{
        //seta os valores de novo e atualiza
        aluno.setNome(nome.getText().toString());
        aluno.setCpf(cpf.getText().toString());
        aluno.setTelefone(telefone.getText().toString());
        dao.atualizar(aluno);
        //MENSAGEM
        Toast.makeText(this, "Aluno foi Atualizado!",
Toast.LENGTH_SHORT).show();
    }
}

// Este método é necessário para garantir a navegação correta para
trás
/*
@Override
public boolean onSupportNavigateUp() {
    NavController navController =
Navigation.findNavController(this,
R.id.nav_host_fragment_activity_main);
    return NavigationUI.navigateUp(navController,
appBarConfiguration)
        || super.onSupportNavigateUp();
}*/

```

