

// Projeto da Faculdade - Organização de
Tarefas

// Desenvolvido em Java (POO simples)

// Este programa permite cadastrar
usuários, equipes, projetos e tarefas

// e listar todas as informações
cadastradas.

import java.util.ArrayList;

import java.util.Scanner;

// Classe Usuario - representa alunos ou
professores

class Usuario {

String nome;

String tipo; // Aluno ou Professor

Usuario(String nome, String tipo) {

```
    this.nome = nome;  
    this.tipo = tipo;  
}
```

```
@Override  
public String toString() {  
    return tipo + ": " + nome;  
}  
}
```

```
// Classe Equipe - agrupa usuários  
class Equipe {  
    String nome;  
    ArrayList<Usuario> membros = new  
    ArrayList<>();
```

```
    Equipe(String nome) {  
        this.nome = nome;  
    }
```

```
    public void adicionarMembro(Usuario
```

```
usuario) {  
    membros.add(usuario);  
}
```

```
@Override  
public String toString() {  
    return "Equipe: " + nome + " |  
Membros: " + membros;  
}  
}
```

```
// Classe Projeto - vinculado a uma equipe  
class Projeto {  
    String titulo;  
    Equipe equipe;  
  
    Projeto(String titulo, Equipe equipe) {  
        this.titulo = titulo;  
        this.equipe = equipe;  
    }  
}
```

```
@Override  
public String toString() {  
    return "Projeto: " + titulo + " | Equipe: "  
+ equipe.nome;  
}  
}
```

// Classe Tarefa - vinculada a um projeto e responsável

```
class Tarefa {  
    String descricao;  
    Projeto projeto;  
    Usuario responsavel;  
  
    Tarefa(String descricao, Projeto projeto,  
Usuario responsavel) {  
        this.descricao = descricao;  
        this.projeto = projeto;  
        this.responsavel = responsavel;  
    }  
}
```

```
@Override  
public String toString() {  
    return "Tarefa: " + descricao +  
        " | Projeto: " + projeto.titulo +  
        " | Responsável: " +  
responsavel.nome;  
}  
}
```

```
// Classe principal  
public class ProjetoFaculdade {  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
  
        // Listas para armazenar os objetos  
criados  
        ArrayList<Usuario> usuarios = new  
ArrayList<>();  
        ArrayList<Equipe> equipes = new  
ArrayList<>();  
        ArrayList<Projeto> projetos = new
```

```
ArrayList<>();  
    ArrayList<Tarefa> tarefas = new  
ArrayList<>();  
  
    int opcao;  
    do {  
        // Menu principal  
        System.out.println("\n=== MENU  
PRINCIPAL ===");  
        System.out.println("1 - Cadastrar  
Usuário");  
        System.out.println("2 - Cadastrar  
Equipe");  
        System.out.println("3 - Cadastrar  
Projeto");  
        System.out.println("4 - Cadastrar  
Tarefa");  
        System.out.println("5 - Listar  
Usuários");  
        System.out.println("6 - Listar  
Equipes");
```

```
        System.out.println("7 - Listar  
Projetos");  
        System.out.println("8 - Listar  
Tarefas");  
        System.out.println("0 - Sair");  
        System.out.print("Escolha uma  
opção: ");  
        opcao = sc.nextInt();  
        sc.nextLine(); // limpar buffer  
  
        switch (opcao) {  
            case 1: // Cadastro de usuário  
                System.out.print("Nome do  
usuário: ");  
                String nome = sc.nextLine();  
                System.out.print("Tipo (Aluno/  
Professor): ");  
                String tipo = sc.nextLine();  
                usuarios.add(new  
Usuario(nome, tipo));  
                System.out.println("Usuário
```

```
cadastrado com sucesso!");  
    break;
```

```
    case 2: // Cadastro de equipe  
        System.out.print("Nome da  
equipe: ");  
        String nomeEquipe =  
sc.nextLine();  
        Equipe equipe = new  
Equipe(nomeEquipe);  
        System.out.println("Adicione  
membros pelo índice:");  
        for (int i = 0; i < usuarios.size();  
i++) {  
            System.out.println(i + " - " +  
usuarios.get(i));  
        }  
        System.out.print("Digite o  
índice do membro (ou -1 para parar): ");  
        int indice;  
        while ((indice = sc.nextInt()) !=
```



```
-1) {  
        if (indice >= 0 && indice <  
usuarios.size()) {  
  
equipe.adicionarMembro(usuarios.get(indi  
ce));  
  
        } else {  
            System.out.println("Índice  
inválido!");  
        }  
        System.out.print("Digite  
outro índice (ou -1 para parar): ");  
    }  
    equipes.add(equipe);  
    System.out.println("Equipe  
cadastrada com sucesso!");  
    break;
```

```
case 3: // Cadastro de projeto  
    if (equipes.isEmpty()) {
```

```
System.out.println("Nenhuma equipe
cadastrada!");
        break;
    }
    System.out.print("Título do
projeto: ");
    String titulo = sc.nextLine();
    System.out.println("Escolha a
equipe pelo índice:");
    for (int i = 0; i < equipes.size(); i+
+) {
        System.out.println(i + " - " +
equipes.get(i).nome);
    }
    int indiceEquipe = sc.nextInt();
    if (indiceEquipe >= 0 &&
indiceEquipe < equipes.size()) {
        projetos.add(new
Projeto(titulo, equipes.get(indiceEquipe)));
        System.out.println("Projeto
cadastrado com sucesso!");
```

```
        } else {  
            System.out.println("Índice  
inválido!");  
        }  
        break;
```

```
    case 4: // Cadastro de tarefa  
        if (projetos.isEmpty() ||  
usuarios.isEmpty()) {  
            System.out.println("Cadastre  
projetos e usuários antes!");  
            break;  
        }  
        System.out.print("Descrição da  
tarefa: ");  
        String descricao = sc.nextLine();  
        System.out.println("Escolha o  
projeto pelo índice:");  
        for (int i = 0; i < projetos.size();  
i++) {  
            System.out.println(i + " - " +
```

```
    projetos.get(i).titulo);
        }
        int indiceProjeto = sc.nextInt();
        System.out.println("Escolha o
responsável pelo índice:");
        for (int i = 0; i < usuarios.size();
i++) {
            System.out.println(i + " - " +
usuarios.get(i));
        }
        int indiceUsuario = sc.nextInt();
        if (indiceProjeto >= 0 &&
indiceProjeto < projetos.size()
            && indiceUsuario >= 0 &&
indiceUsuario < usuarios.size()) {
            tarefas.add(new
Tarefa(descricao,
projetos.get(indiceProjeto),
usuarios.get(indiceUsuario)));
            System.out.println("Tarefa
cadastrada com sucesso!");
```

```
    } else {  
        System.out.println("Índices  
inválidos!");  
    }  
    sc.nextLine(); // limpar buffer  
    break;
```

```
case 5: // Listar usuários  
    System.out.println("\n===  
Usuários ===");  
    for (Usuario u : usuarios) {  
        System.out.println(u);  
    }  
    break;
```

```
case 6: // Listar equipes  
    System.out.println("\n===  
Equipes ===");  
    for (Equipe eq : equipes) {  
        System.out.println(eq);  
    }
```

```
break;
```

```
case 7: // Listar projetos
```

```
    System.out.println("\n===  
Projetos ===");  
    for (Projeto p : projetos) {  
        System.out.println(p);  
    }  
    break;
```

```
case 8: // Listar tarefas
```

```
    System.out.println("\n===  
Tarefas ===");  
    for (Tarefa t : tarefas) {  
        System.out.println(t);  
    }  
    break;
```

```
case 0: // Sair
```

```
    System.out.println("Saindo do  
sistema...");
```

```
break;
```

```
default:
```

```
    System.out.println("Opção  
inválida!");
```

```
    }
```

```
} while (opcao != 0);
```

```
sc.close();
```

```
}
```

```
}
```