

Caio Dantas Farias 24026710

Kauã Daniel de Aguiar Ferreira 24026846

Saulo Ribeiro Santos 24026911

## Classes Para Recebimento e Envio de Dados Programados

Com base no que foi pedido no projeto, na entrega, e na disciplina de Programação Orientada a Objetos, a classe escolhida como principal foi “Dashboard” que representa a estrutura central do sistema. Esta classe é a base para todas as outras funcionalidades do dashboard e para as outras classes através da herança. As classes que irão herdar da classe Dashboard serão: Evento, Projeto, Tarefa e Notificação. Cada uma dessas classes irá acrescentar atributos e funcionalidades específicas, porém mantendo a estrutura geral do Dashboard.

O código foi feito com base no que foi passado em sala de aula, utilizando alguns aprendizados como gets e sets manuais, funções construtoras e herança.

Também foi feito na linguagem C# pois é a que estamos mais familiarizados por conta das aulas, porém podendo ser atualizada para Javascript conforme a evolução do projeto.

```
Dashboard.cs > Dashboard
1 // Classe principal
2 public class Dashboard {
3     private int Id;
4     private string Nome;
5
6     public int GetId() {
7         return Id;
8     }
9     public void SetId(int novoId) {
10        Id = novoId;
11    }
12
13    public string GetNome() {
14        return Nome;
15    }
16    public void SetNome(string novoNome) {
17        Nome = novoNome;
18    }
19
20    public Dashboard() { }
21
22    public Dashboard(int id) {
23        Id = id;
24        Nome = "Sem nome";
25    }
26
27    public Dashboard(int id, string nome) {
28        Id = id;
29        Nome = nome;
30    }
31
32    public string ExibirNome() {
33        return Nome;
34    }
35 }
```

Dashboard.cs

```
Eventos.cs > Evento
1 // Classe Evento (herança de Dashboard)
2 public class Evento : Dashboard {
3     private string Local;
4     private string Descricao;
5     private string DataEvento;
6     private string Status;
7
8     public string GetLocal() {
9         return Local;
10    }
11    public void SetLocal(string novoLocal) {
12        Local = novoLocal;
13    }
14
15    public string GetDescricao() {
16        return Descricao;
17    }
18    public void SetDescricao(string novaDescricao) {
19        Descricao = novaDescricao;
20    }
21
22    public string GetDataEvento() {
23        return DataEvento;
24    }
25    public void SetDataEvento(string novaData) {
26        DataEvento = novaData;
27    }
28
29    public string GetStatus() {
30        return Status;
31    }
32    public void SetStatus(string novoStatus) {
33        Status = novoStatus;
34    }
35
36    public Evento() { }
```

```
Dashboard.cs | Eventos.cs X | Projeto.cs | Tarefa.cs | Notificação.cs
Eventos.cs > Evento
2 public class Evento : Dashboard {
3     public Evento() { }
4
5     public Evento(int id, string nome) : base(id, nome) {
6         Local = "Local indefinido";
7         Descricao = "Sem descrição";
8         DataEvento = "Data não definida";
9         Status = "Indefinido";
10    }
11
12    public Evento(int id, string nome, string local, string descricao, string dataEvento, string status) : base(id, nome) {
13        Local = local;
14        Descricao = descricao;
15        DataEvento = dataEvento;
16        Status = status;
17    }
18 }
```

Evento.cs

Tarefa.cs; Notificação.cs; Projeto.cs

```

Tarefas > Tarefa
1 // Classe Tarefa (herança de Dashboard)
2 references
3 public class Tarefa : Dashboard {
4     3 references
5     private string Descricao;
6     3 references
7     private string DataConclusao;
8     3 references
9     private string Status;
10
11     0 references
12     public string GetDescricao() {
13         return Descricao;
14     }
15     0 references
16     public void SetDescricao(string novaDescricao) {
17         Descricao = novaDescricao;
18     }
19
20     0 references
21     public string GetDataConclusao() {
22         return DataConclusao;
23     }
24     0 references
25     public void SetDataConclusao(string novaDataConclusao) {
26         DataConclusao = novaDataConclusao;
27     }
28
29     0 references
30     public string GetStatus() {
31         return Status;
32     }
33     0 references
34     public void SetStatus(string novoStatus) {
35         Status = novoStatus;
36     }
37
38     0 references
39     public Tarefa() { }
40
41     0 references
42     public Tarefa(int id, string nome, string descricao, string dataConclusao, string status) : base(id, nome) {
43         Descricao = descricao;
44         DataConclusao = dataConclusao;
45         Status = status;
46     }
47 }

```

```

Notificações > Notificacao
1 // Classe Notificacao (herança de Dashboard)
2 references
3 public class Notificacao : Dashboard {
4     3 references
5     private string Mensagem;
6
7     0 references
8     public string GetMensagem() {
9         return Mensagem;
10    }
11    0 references
12    public void SetMensagem(string novaMensagem) {
13        Mensagem = novaMensagem;
14    }
15
16    0 references
17    public Notificacao() { }
18
19    0 references
20    public Notificacao(int id, string nome, string mensagem) : base(id, nome) {
21        Mensagem = mensagem;
22    }
23 }

```

```

Projeto.cs > Projeto
1 // Classe Projeto (herança de Dashboard)
2 references
3 public class Projeto : Dashboard
4 {
5     3 references
6     private string Descricao;
7     3 references
8     private string DataInicio;
9     3 references
10    private string DataFim;
11
12    0 references
13    public string GetDescricao() {
14        return Descricao;
15    }
16
17    0 references
18    public void SetDescricao(string novaDescricao) {
19        Descricao = novaDescricao;
20    }
21
22    0 references
23    public string GetDataInicio() {
24        return DataInicio;
25    }
26
27    0 references
28    public void SetDataInicio(string novaDataInicio) {
29        DataInicio = novaDataInicio;
30    }
31
32    0 references
33    public string GetDataFim() {
34        return DataFim;
35    }
36
37    0 references
38    public void SetDataFim(string novaDataFim) {
39        DataFim = novaDataFim;
40    }
41
42    0 references
43    public Projeto() { }
44
45    0 references
46    public Projeto(int id, string nome, string descricao, string dataInicio, string dataFim) : base(id, nome) {
47        Descricao = descricao;
48        DataInicio = dataInicio;
49        DataFim = dataFim;
50    }
51 }

```