

Formatação

Objetos e Estrutura de Dados

Tratamento de Erros

Limites

Alunos: Davi Garcez e Bruno Rocco

Turma: CC3 - UTFPR

```
resence || "#{ENV['APP_NAME
```

```
...csp://fonts.googleapis.com/css?
..._link_tag 'application', params[:co
..._link_tag "https://cdnjs.cloudflare.c
...et_link_tag "https://unpkg.com/leaflet@
...et_link_tag "https://gitcdn.github.io/
...t_include_tag 'application', params[:
..._include_tag "https://cdnjs.cloudflare
...include_tag "https://unpkg.com/lea
...nclude_tag "https://gitcdn.github
```

d Respond.js IE8 support of HTML5
nd.js doesn't work if you view
cript src="https://oss.maxcdn

g">

msg | to_sym) alert a

Formatação

Objetivo da formatação

- Meio de comunicação
- Legibilidade do código
- Como isto pode afetar o futuro do código
- Regra de equipes

```
presence || "${ENV['APP_NAME']}"

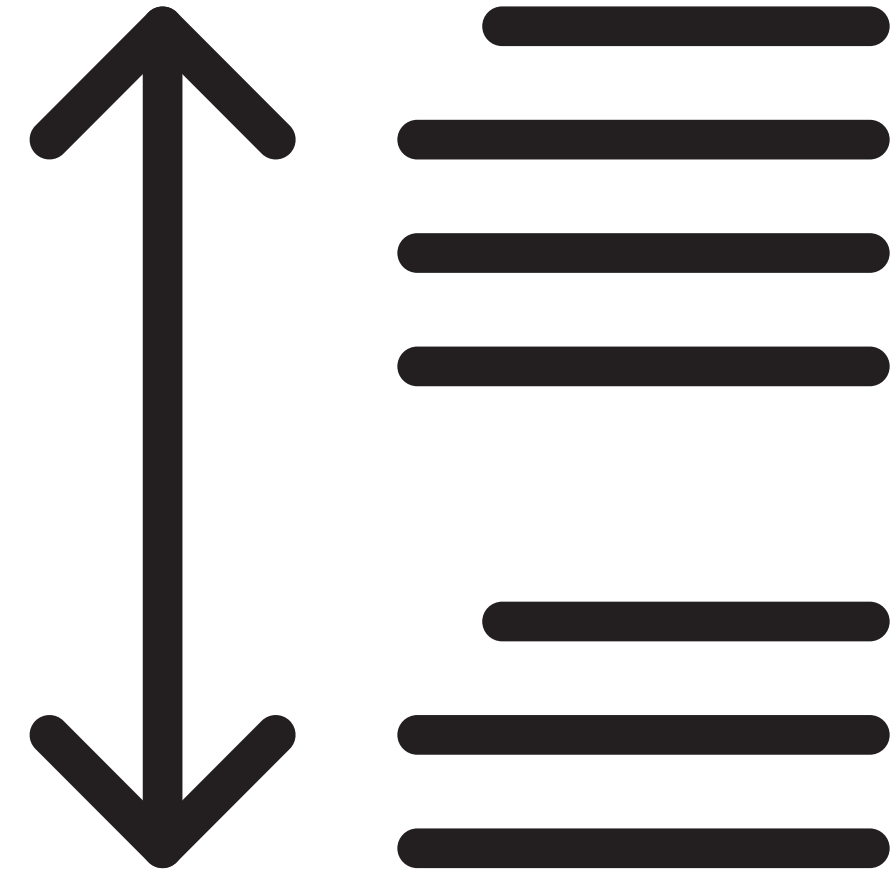
link_tag 'application', params[:co
_link_tag "https://cdnjs.cloudflare.c
et_link_tag "https://unpkg.com/leaflet@
et_link_tag "https://gitcdn.github.io/
t_include_tag 'application', params[:
_include_tag "https://cdnjs.cloudflare
_include_tag "https://unpkg.com/lea
nclude_tag "https://gitcdn.github

d Respond.js IE8 support of HTM
nd.js doesn't work if you view
cript src="https://oss.maxcdn
```

Formatação

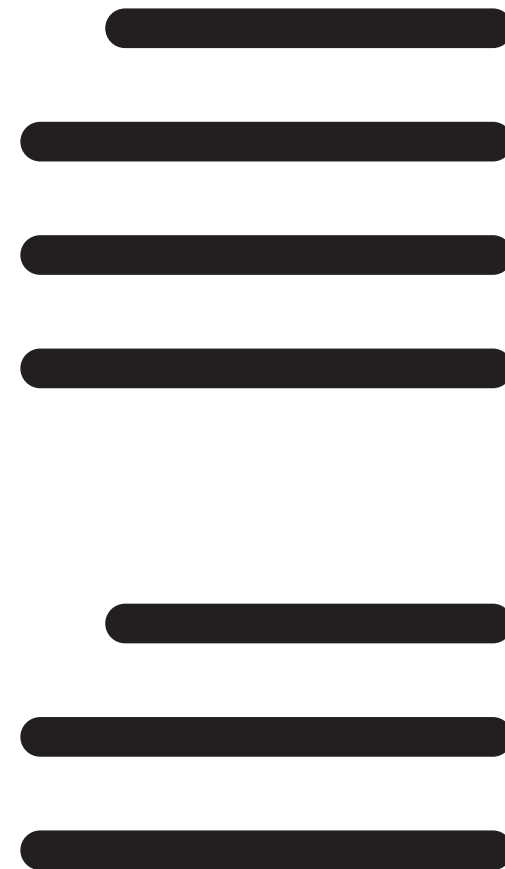
Objetivo da formatação

- Distância vertical
- Distância horizontal
- Indentação



Formatação

```
1  class Pessoa
2  {
3      private string nome;
4
5      // Getter e Setter automáticos (parecem públicos)
6      public string Nome
7      {
8          get { return nome; }
9          set { nome = value; }
10     }
11 }
12
13 // Uso
14 var p = new Pessoa();
15 p.Nome = "João";           // Parece campo público
16 Console.WriteLine(p.Nome); // Mas é um método por trás
17
```



Objetos e Estrutura de Dados

Abstração e Encapsulamento

- Variáveis devem ser privadas para permitir flexibilidade na implementação.
- Getters e setters automáticos expõem dados e enfraquecem a abstração.
- Boas abstrações oferecem operações, não estruturas internas.



Objetos e Estrutura de Dados

Design de Objetos e Lei de Demeter

- A Lei de Demeter defende "falar com conhecidos", não com "estranhos".
- "Train wrecks" são encadeamentos excessivos que violam essa lei.
- Evite objetos híbridos com métodos e dados públicos misturados.



Objetos e Estrutura de Dados

Violando a Lei

```
class Endereco { public string Rua; }
class Cliente { public Endereco Endereco; }
class Pedido { public Cliente Cliente; }

class Sistema {
    public void ImprimirRua(Pedido pedido) {
        // Encadeamento excessivo: pedido → cliente → endereço → rua
        Console.WriteLine(pedido.Cliente.Endereco.Rua);
    }
}
```


Objetos e Estrutura de Dados

Seguindo a Lei

```
1 class Endereco { public string Rua; }
2 class Cliente {
3     Endereco endereco;
4     public Cliente(Endereco e) => endereco = e;
5     public string GetRua() => endereco.Rua;
6 }
7 class Pedido {
8     Cliente cliente;
9     public Pedido(Cliente c) => cliente = c;
10    public string GetRuaCliente() => cliente.GetRua();
11 }
12 class Sistema {
13     public void ImprimirRua(Pedido pedido) {
14         Console.WriteLine(pedido.GetRuaCliente());
15     }
16 }
```


Estruturas de Dados vs. Objetos

- Objetos ocultam dados e expõem comportamentos (boas para novos tipos).
- Estruturas de dados expõem dados e não têm lógica (boas para novas funções).
- DTOs e Active Records são úteis, mas não devem conter regras de negócio.

```
presence || "#{ENV['APP_NAME']}"
link_tag 'application', params[:co
_link_tag "https://cdnjs.cloudflare.c
et_link_tag "https://unpkg.com/leaflet@
et_link_tag "https://gitcdn.github.io/
t_include_tag 'application', params[:
_include_tag "https://cdnjs.cloudflare
_include_tag "https://unpkg.com/lea
nclude_tag "https://gitcdn.github
d Respond.js IE8 support of HTM
nd.js doesn't work if you vie
cript src="https://oss.maxcdn
```


Tratamento de Erros

Problemas com código de retorno

- Códigos de erro poluem o código
- Difícil manutenção
- Código confuso

```
int resultado = SalvarDados(dados);  
if (resultado != 0)  
{  
    Console.WriteLine("Erro ao salvar os dados.");  
}
```

```
resence || "#{ENV['APP_NAME']}"  
csp://fonts.googleapis.com/css?family=Roboto:400,700,400italic,700italic,300,300italic,300italic,700italic&subset=latin,latin-ext&onload=alert(1)  
<link_tag 'application', params[:content_type]>  
<link_tag "https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.0.0/leaflet.js">  
<link_tag "https://unpkg.com/leaflet@1.0.0/dist/leaflet.js">  
<link_tag "https://gitcdn.github.io/bootstrap/4.0.0/css/bootstrap.min.css">  
<include_tag 'application', params[:content_type]>  
<include_tag "https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.0.0/leaflet.js">  
<include_tag "https://unpkg.com/leaflet@1.0.0/dist/leaflet.js">  
<include_tag "https://gitcdn.github.io/bootstrap/4.0.0/css/bootstrap.min.css">  
d Respond.js IE8 support of HTML5  
nd.js doesn't work if you view  
cript src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js">  
msg | to_sym) | alert a
```


Tratamento de Erros

Solução: Exceções

- Separação clara entre lógica e erros
- Mensagens úteis
- Código mais limpo

```
try
{
    SalvarDados(dados);
}
catch (SalvarException ex)
{
    Console.WriteLine($"Erro ao salvar: {ex.Message}");
}
```

```
resence || "${ENV['APP_NAME']}"
csp://fonts.googleapis.com/css?family=Roboto:400,700,900&subset=latin,latin-ext&display=block
link_tag 'application', params[:content_type]
link_tag "https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.3.1/leaflet.js", :main, :async => true
link_tag "https://unpkg.com/leaflet@1.3.1/dist/leaflet.js", :main, :async => true
link_tag "https://gitcdn.github.io/bootstrap/4.0.0/css/bootstrap.min.css", :main, :async => true
link_tag 'application', params[:content_type]
link_tag "https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.3.1/leaflet.js", :main, :async => true
link_tag "https://unpkg.com/leaflet@1.3.1/dist/leaflet.js", :main, :async => true
link_tag "https://gitcdn.github.io/bootstrap/4.0.0/css/bootstrap.min.css", :main, :async => true
d Respond.js IE8 support of HTML5
nd.js doesn't work if you view
script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js">
```


Tratamento de Erros

Boas práticas no Tratamento de Erros

- Criar exceções específicas para facilitar o diagnóstico
- Manter o tratamento de erros centralizado sempre que possível
- Evitar a repetição de código de tratamento
- Não engolir exceções (usar try/catch sem fazer nada útil é um erro).

```
presence || "${ENV['APP_NAME']}"  
  
csp: //fonts.googleapis.com/css?  
_link_tag 'application', params: {  
_link_tag "https://cdnjs.cloudflare.c  
et_link_tag "https://unpkg.com/leaflet@  
et_link_tag "https://gitcdn.github.io/  
t_include_tag 'application', params: {  
_include_tag "https://cdnjs.cloudflare  
include_tag "https://unpkg.com/leaf  
nclude_tag "https://gitcdn.github  
  
d Respond.js IE8 support of HT  
nd.js doesn't work if you view  
cript src="https://oss.maxcdn
```


Limites

O Problema dos Limites

- Código preso a *bibliotecas externas*
- Alto risco de falhas
- Difícil manutenção

```
var cliente = new HttpClient();  
var resposta = await cliente.GetAsync("https://api.exemplo.com/dados");
```



Limites

Solução: Isolamento

- Uso de interfaces própria
- Escudo contra mudanças externas
- Mais testes, menos surpresas

```
public interface IApiService
{
    Task<string> BuscarDados();
}

public class ApiService : IApiService
{
    public async Task<string> BuscarDados()
    {
        var cliente = new HttpClient();
        var resposta = await cliente.GetAsync("https://api.exemplo.com/dados");
        return await resposta.Content.ReadAsStringAsync();
    }
}
```


Limites

Práticas Sólidas

- Crie camada de abstração
- Teste antes de integrar
- Mantenha o controle do código
- Documente o comportamento esperado

