

EMBARCADERO CONFERENCE



embarcadero®

Evitando memory leaks e tratando exceções

Maico Dal Ri

RAD

CONFERENCE
EMBARCADERO CONFERENCE 2022

Evitando memory leaks e tratando exceções

Quem sou eu?

- Desenvolvedor Delphi há mais de uma década.
- Trabalho como Outsourcing na Aquasoft há mais de 6 anos.
- Certificado Developer e Master Developer pela Embarcadero.



Evitando memory leaks e tratando exceções

O que vamos ver hoje?



- O Sistema Operacional e a Memória do seu aplicativo
- O que são Variáveis, Instâncias de um Objeto ou Classe?
- O que é sujeira de memória?
- O que é um Access Violation?
- O que é Memory Leaks?
- O que é uma Exceção?
- O Application OnException.



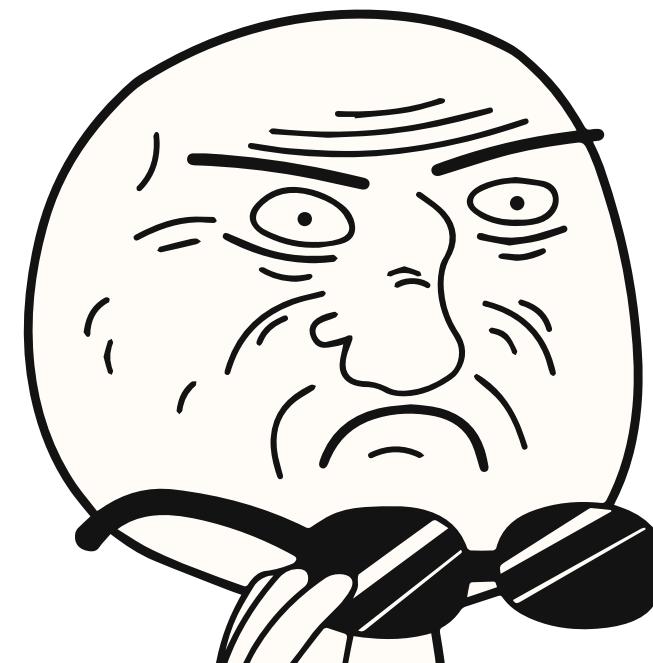
Evitando memory leaks e tratando exceções

O Sistema Operacional e a Memória do seu aplicativo

Evitando memory leaks e tratando exceções

O Sistema Operacional e a Memória do seu aplicativo

Quem gerencia a memória
física do equipamento?

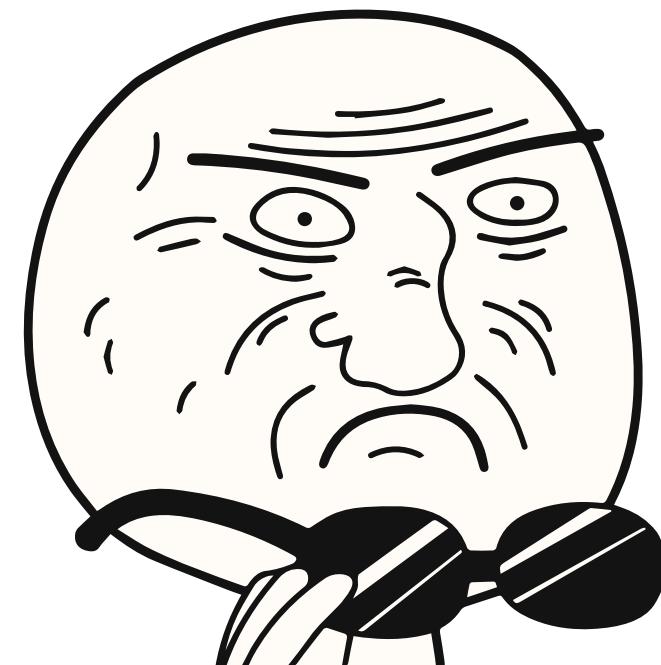


LADO BOM

Evitando memory leaks e tratando exceções

O Sistema Operacional e a Memória do seu aplicativo

Quem gerencia a memória física do equipamento?

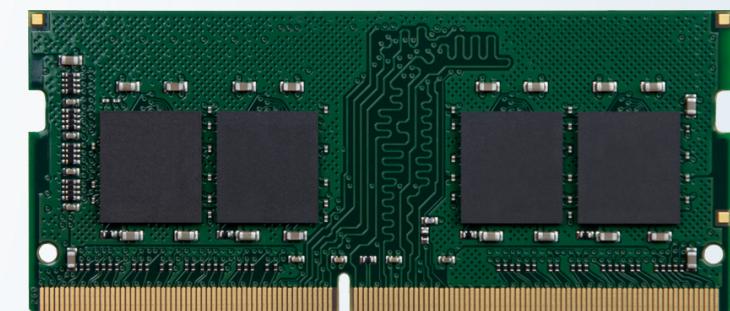


UMA DAS RESPONSABILIDADES PRINCIPAIS DO SO É GERENCIAR A MEMÓRIA FÍSICA!

Evitando memory leaks e tratando exceções

O que são variáveis, instâncias de uma classe ou objetos?

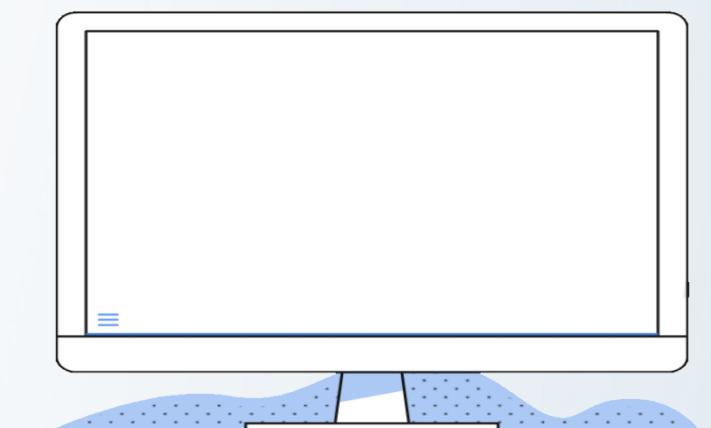
Variáveis nada mais são do que apelidos que pedimos ao compilador para referenciar determinado segmento de memória.



Evitando memory leaks e tratando exceções

O que são variáveis, instâncias de uma classe ou objetos?

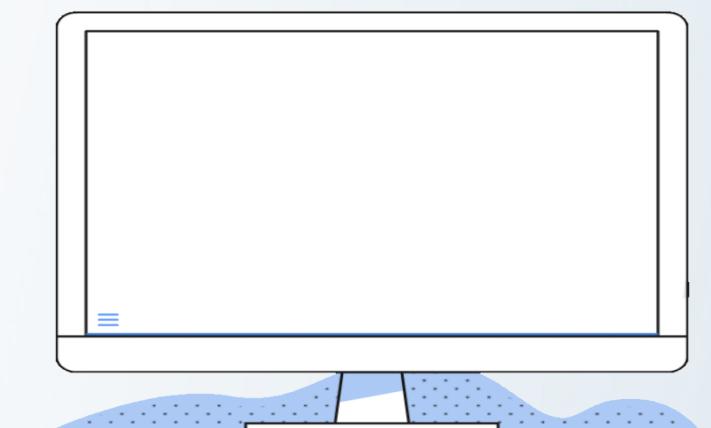
Objetos ou instâncias de classes são apelidos de segmentos de memória, porém com estruturas bem mais complexas que variáveis, códigos que devem ser executados pelo processador.



Evitando memory leaks e tratando exceções

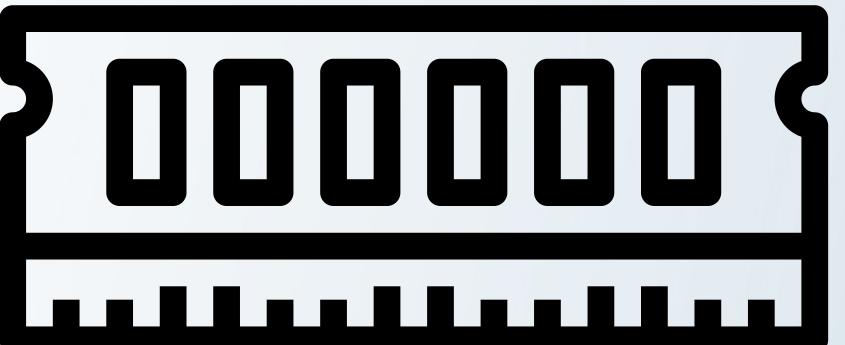
O que são variáveis, instâncias de uma classe ou objetos?

Objetos ou instâncias de classes são apelidos de segmentos de memória, porém com estruturas bem mais complexas que variáveis, códigos que devem ser executados pelo processador.



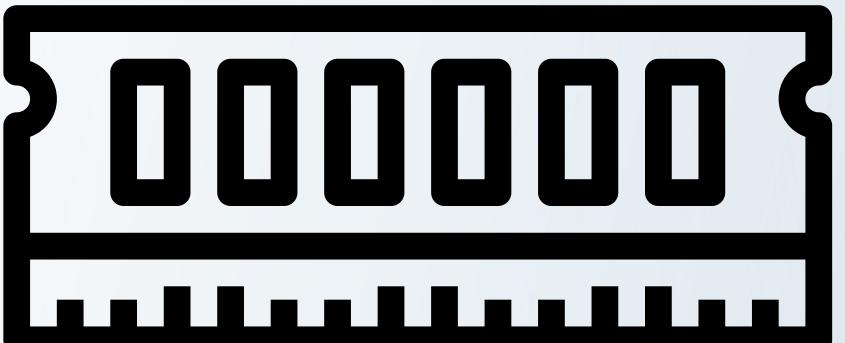
Evitando memory leaks e tratando exceções

1.Uma variável é uma referência da memória!



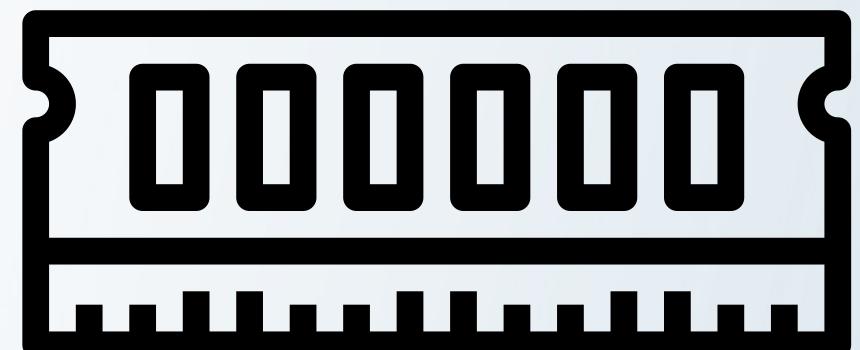
Evitando memory leaks e tratando exceções

- 1.Uma variável é uma referência da memória!
- 2.Quando criamos um objeto, preenchemos o conteúdo da memória com a estrutura do objeto.



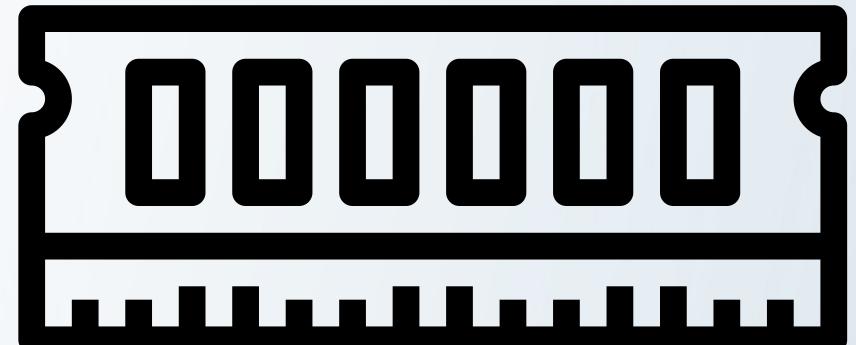
Evitando memory leaks e tratando exceções

- 1.Uma variável é uma referência da memória!
- 2.Quando criamos um objeto, preenchemos o conteúdo da memória com a estrutura do objeto.
- 3.Quando utilizamos um método de um objeto, estamos acessando um determinado ponto da estrutura daquele objeto que está localizado na memória indicada pela variável associada ao objeto.



Evitando memory leaks e tratando exceções

- 1.Uma variável é uma referência da memória!
- 2.Quando criamos um objeto, preenchemos o conteúdo da memória com a estrutura do objeto.
- 3.Quando utilizamos um método de um objeto, estamos acessando um determinado ponto da estrutura daquele objeto que está localizado na memória indicada pela variável associada ao objeto.
- 4.Destruir o objeto não altera em nada sua referência.



Evitando memory leaks e tratando exceções

O que é sujeira de memória?

É quando deixamos um segmento de memória reservado, porém sem informações válidas.



```
var
  lStringList: TStringList; //Declaração do objeto
  {Basicamente, estamos reservando um espaço de memória
  para conter o objeto em questão}
begin
  lStringList:= TStringList.Create; {Preenchemos este espaço de memória
  com informações e procedimentos
  do referido objeto}
  //Usamos o Objeto
  lStringList.Free; {Destruímos o objeto.
  Na prática, apenas limpamos o objeto da memória e
  devolvemos a memória para o SO}
end;
```

Evitando memory leaks e tratando exceções

Variáveis e seus lixos!



**"TODA VARIÁVEL
NÃO
INICIALIZADA,
NATURALMENTE
POSSUI LIXO."**

Maico Dal Ri.



Evitando memory leaks e tratando exceções

Variáveis e seus lixos!



**"TODA VARIÁVEL
NÃO
INICIALIZADA,
NATURALMENTE
POSSUI LIXO."**

Maico Dal Ri.

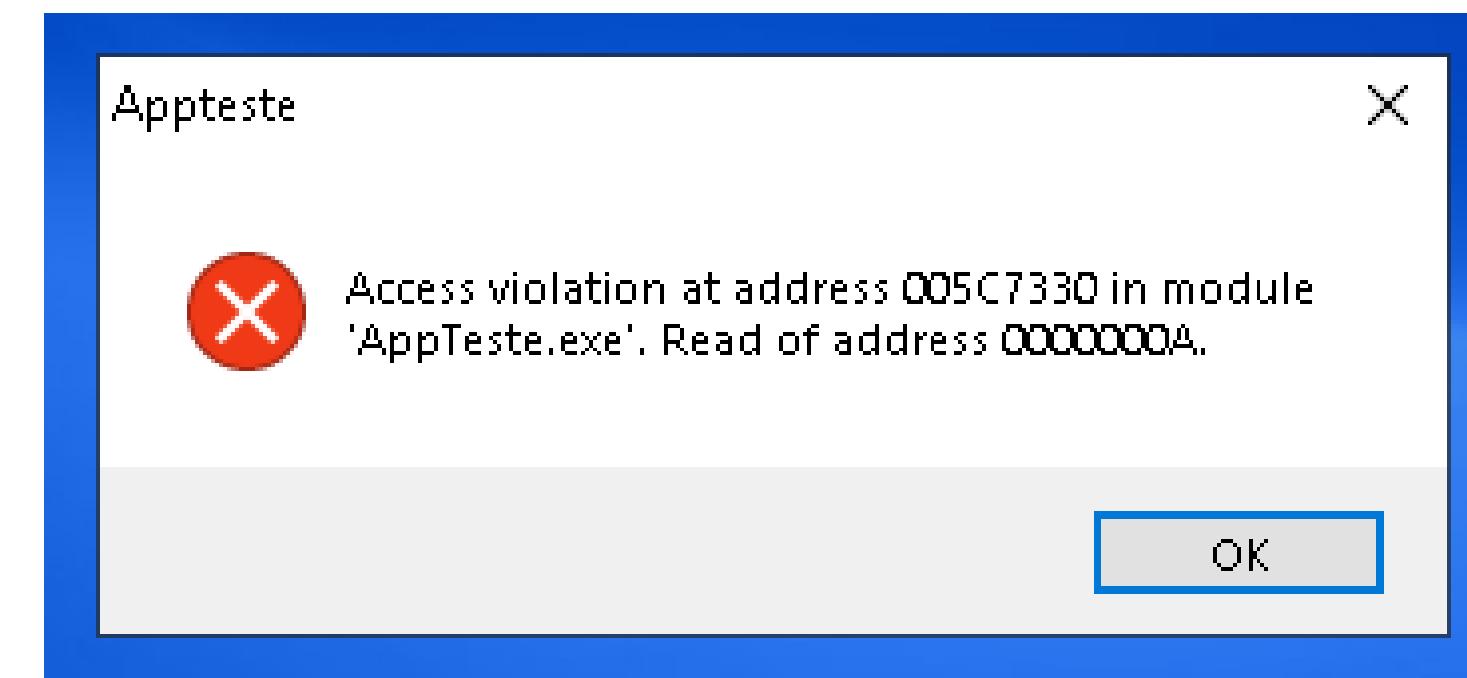
**EM OBJECT PASCAL (DELPHI), A ÚNICA VARIÁVEL QUE É
AUTOMATICAMENTE INICIALIZADA É A STRING!**

Evitando memory leaks e tratando exceções

O que é um Access Violation?

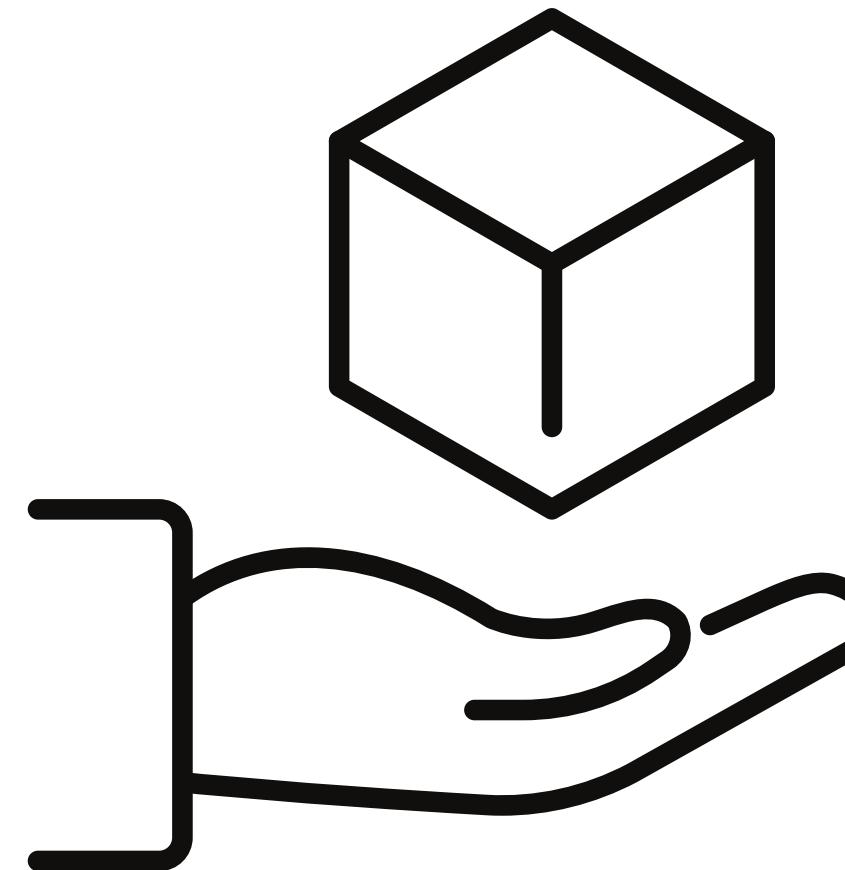
Tradução: "Violação de acesso!" ou "Acesso Violento".

Nada mais é do que quando tentamos acessar, por qualquer motivo, um segmento de memória que não pertence a nossa aplicação.



Evitando memory leaks e tratando exceções

Memory Leaks



É quando alocamos e criamos determinado objeto na memória, porém não devolvemos os recursos novamente para o Sistema Operacional!

Evitando memory leaks e tratando exceções

O que é uma Exceção?



Uma exceção é um sinal que indica que algum tipo de condição excepcional ocorreu durante a execução do programa.

Assim, exceções estão associadas a condições de erro que não tinham como ser verificadas durante a compilação do programa.

Evitando memory leaks e tratando exceções

Application.OnException

Ocorre sempre antes de ser exposta ao usuário.

Assinatura:

```
TExceptionEvent = procedure (Sender: TObject; E: Exception) of object;
```

Evitando memory leaks e tratando exceções



Evitando memory leaks e tratando exceções

Perguntas?



OBRIGADO



Maico.dalri@aquasoft.com.br



[http://taurasdodelphi.blogspot.com/?
m=1](http://taurasdodelphi.blogspot.com/?m=1)

