

Relato de testes com IA (22/02/24)

Resumo

- Utilizado nova [URL](#)
- O formato do retorno da API **complete** ficou bom e fácil de usar`
- Parece que o IA está conseguindo detectar o estilo (orientado a objetos ou comandos) e apresentar sugestões coerentes com o estilo.
- Parece que o IA está conseguindo **aprender e evoluir** a cada interação.
- Eliminado efeito de colocar/tirar espaços (na versão anterior, ao adicionar/retirar um espaço, as sugestões eram diferentes).
- Em relação a versão anterior:
 - Retorno da API **complete** limpo, ou seja, sem `\n` em excesso e padronizado;
 - Sugestões seguindo o mesmo estilo e padrões de código do código existente;
 - Sem truncamento de linhas (limite de *tokens* na anterior)

Comentários

- Padronizar o retorno das API (sempre JSON). A **health** retorna texto bruto.
Sugiro colocar também algo como "em manutenção", com previsão de término, como uma das resposta desta API.
- ~~Retorno aparenta estar mais rápido.~~
~~Na versão anterior não fiz medições, portanto é sentimento e passarei a fazê-las, nesta versão e próximas.~~
- Após análise mais cuidadosa, devido ao uso de **Promise** no processo de comunicação com a IA que mascara o tempo real, os tempos de execução da API **complete** ficam entre 15 e 20 seg, com acesso via WI-FI. Via cabo, entre 10s e 13 seg.

Destaques

Respeito ao *estilo*

```
157 |
158 | define msdialog oDlg title "Escolha um número" from 0,0 to 250,400 of oMainWnd PIXEL
    | @ 10,28 SAY "Escolha um número acionando um dos botões abaixo." of oDlg PIXEL
    | @ 90,10 BUTTON "<nenhum>" SIZE 90,20 ACTION (cOpcao := "", oDlg:end()) of oDlg PIXEL
    | @ 90,110 BUTTON "<encerrar>" SIZE 90,20 ACTION (cOp
159 |
```

```

81 oDlg := MSDIALOG():Create()
82 oDlg:cCaption := "Escolha um número"
83 oDlg:cName := "oDlg"
84 oDlg:lBorder := .T.
85 oDlg:lCaption := .T.
86 oDlg:lCentered := .T.
   oDlg:nWidth := 300
   oDlg:nHeight := 150
   oDlg:nLeft := 10
   oDlg:nTop := 10

```

Código comentados estão sendo ignorado (ou não?)

```

oSay1 := TSAY():Create(oDlg)
oSay1:cCaption := "Escolha um número acionando um dos botões abaixo."
oSay1:nLeft := 10
oSay1:nTop := 28
oSay1:nWidth := 250
oSay1:nHeight := 17

// oSay1:cName := "oSay1"
// oSay1:xxxxxxxxx := "Escolha um número acionando um dos botões abaixo."
// oSay1:nLeft := 10
// oSay1:nTop := 28
// oSay1:nWidth := 250

```

NO TREINAMENTO foi usado um fonte de nome `escolheNUm.prw`?

ESTOU SURPRESO:

Código comentado está sendo considerado pelo IA e **corrigindo** (aparenta);

Ou está usando alguma tipo de "memória", pois utilizei esse fonte diversas vezes;

Ou inspirou-se no código OOP que esta nesse mesmo fonte.

Parece que o IA está conseguindo **aprender** e evoluir a cada interação.

Gerando *codeblock*

```

1 aEval(aaOpcoes, [|x,i| ;
2   oBtn := TButton():Create(oDlg),;
3   oBtn:cCaption := x,;
   oBtn:blClicked := &("{| conout('Foi acionado "+x+"'),cOpcao := '"+x+"', oDlg:end() }"),;
   oBtn:nWidth := 30,;
   oBtn:nTop := 60,;
   oBtn:nLeft := (10 * i) + (oBtn:nWidth*(i-1));
4 |)
5

```

Escrito

Gerado

```
aEval(aaOpcoes, { \|x,i\|;
```

Escrito	Gerado
<code>oBtn := TButton():Create(oDlg),;</code>	
<code>oBtn:cCaption := x,;</code>	<code>oBtn:cCaption := x,;</code>
<code>oBtn:blClicked := &("{\\ \\ conout('Foi acionado "+x+"'),cOpcao := '"+x+"', oDlg:end() }"),;</code>	<code>oBtn:blClicked := &("{\\ \\ conout('Foi acionado "+x+"'),cOpcao := '"+x+"', oDlg:end() }"),;</code>
<code>oBtn:nWidth := 30,;</code>	<code>oBtn:nWidth := 30,;</code>
<code>oBtn:nTop := 60,;</code>	<code>oBtn:nTop := 60,;</code>
<code>oBtn:nLeft := (10 *i) + (oBtn:nWidth*(i-1));</code>	<code>oBtn:nLeft := (10 *i) + (oBtn:nWidth*(i-1));</code>
<code>})</code>	<code>})</code>

SURPRESO: O código é o mesmo. Aparenta ter reaproveitado o código existente na função que usa OOP (mesmo fonte).

Removi a parte com OOP e refiz a operação para gerar o *codeblock*. Não foi o mesmo, mas é útil nesse contexto, necessitando ajustar o que esta destacado.

```
aEval(aaOpcoes, { |x,i| ;
  oBtn := TButton():Create(oDlg),;
  oBtn:cCaption := x,;
  oBtn:blClicked := {|| cOpcao := x, **oDlg:end()** },;
  oBtn:nWidth := 90,;
  **oBtn:nTop** := 90 + (i * 30),;
  **oBtn:nLeft** := 10;
})
```

O IA conseguiu identificar o padrão de código (geração de botões dinâmicos) e utilizar os métodos corretamente e a sintaxe (continuação de linha), para gerar o código do *codeblock*.

E apresentou outras sugestões, variando o tamanho e posição dos botões.

Mais propriedades

No *dialog*, foi apresentado sugestão de propriedades (aparentam estar corretas).

```
81 oDlg := MSDIALOG():Create()
82 oDlg:cCaption := "Escolha um número"
83 oDlg:cName := "oDlg"
84 oDlg:lBorder :=.T.
85 oDlg:lCaption :=.T.
86 oDlg:nWidth := 300
   oDlg:nHeight := 150
   oDlg:nLeft := 0
   oDlg:nTop := 0

87 oSay1 := TSAY():Create(oDlg)
88
89 // oSay1:cName := "oSay1"
90 // oSay1:xxxxxxx := "Escolha um número acionando um dos botões abaixo "
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL TDS-Dito

```
2024-02-22 17:04:52.491 [trace] textRequest (9509 ms)
2024-02-22 17:04:52.493 [debug] Code completions end with 3 suggestions
2024-02-22 17:04:52.493 [debug] [
  {
    "generated_text": "oDlg:nWidth := 300\n\toDlg:nHeight := 150\n\toDlg:nLeft := 0\n\toDlg:nTop := 0\n"
  },
  {
    "generated_text": "oDlg:nWidth := 280\n\toDlg:nHeight := 150\n\toDlg:nLeft := 5\n\toDlg:nTop := 5\n\toDlg:lMaximized :=.T.\n"
  },
  {
    "generated_text": "oDlg:lCentered :=.T.\n\toDlg:nLeft := 0\n\toDlg:nTop := 0\n\toDlg:nWidth := 300\n\toDlg:nHeight := 220\n"
  }
]
```

Fiquei acionando o **tab** para aceitar e continuar solicitando novas sugestões, que foi atendido prontamente com mais propriedades. Repeti o ciclo várias vezes, até que começou a apresentar repetições (propriedades já atribuídas).

```
95 oDlg:lVisible :=.T.
96 oDlg:lCentered :=.T.
97 oDlg:lCentered :=.T.
98 oDlg:lCentered :=.T.
99 oDlg:lCentered :=.T.
   oDlg:lCentered :=.T.
   oDlg:lCentered :=.T.
   oDlg:lCentered :=.T.
   oDlg:lCentered :=.T.
```

No caso do `button`, isso não ocorreu.

```
103
104     oBtn := TButton():Create(oDlg)
105     oBtn:cCaption := "<encerrar>"
106     oBtn:blClicked := {|| cOpcao := "*", oDlg:end() }
107     oBtn:nWidth := 90
108     oBtn:nTop := 90
109     oBtn:nLeft := 110
110
111     aEval(aaOpcoes, { |x,i| ;
112         oBtn := TButton():Create(oDlg),;
113         oBtn:cCaption := x,;
114         oBtn:blClicked := {|| cOpcao := x, oDlg:end() } .
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
2024-02-22 17:02:38.885 [trace] textRequest (2927 ms)
2024-02-22 17:02:38.887 [debug] Code completions end with 3 suggestions
2024-02-22 17:02:38.887 [debug] [
  {
    "generated_text": "\n"
  },
  {
    "generated_text": "\n"
  },
  {
    "generated_text": "\n"
  }
]
```

@..say

No comando `@..say`, especificamente, as sugestões apresentas são mais simples em relação a versão anterior.

Anterior:

```
75
76     local meuDlg
77
78     @ 10,1 get oSay1 from "Label 01" SIZE 100,100 PIXEL
79     @ 10,10 get oSay2 from "Label 02" SIZE 100,100 PIXEL
80     @ 10,20 get oSay3 from "Label 03" SIZE 100,100 PIXEL
81
82     @ 20,1 get oSay4 from "Label 04" SIZE 100,100 PIXEL
83     @ 20,10 get oSay5 from "
84
85     ACTIVATE DIALOG meuDlg CENTERED
86
87     Return
88
```

Atual:

```
9
10 @ 10,01 say "teste"
    @ 10,01 say "teste"
    @ 10,01 say "teste"
    @ 10,01 say "teste"
    @ 10,01 say "teste"
11 if replay == "true"
```

Refinamento



Definição classe

Após indicar fim de classe (`end class`), uma das sugestões foi a prototipação de métodos.