

FIPP – Faculdade de Informática de Presidente Prudente

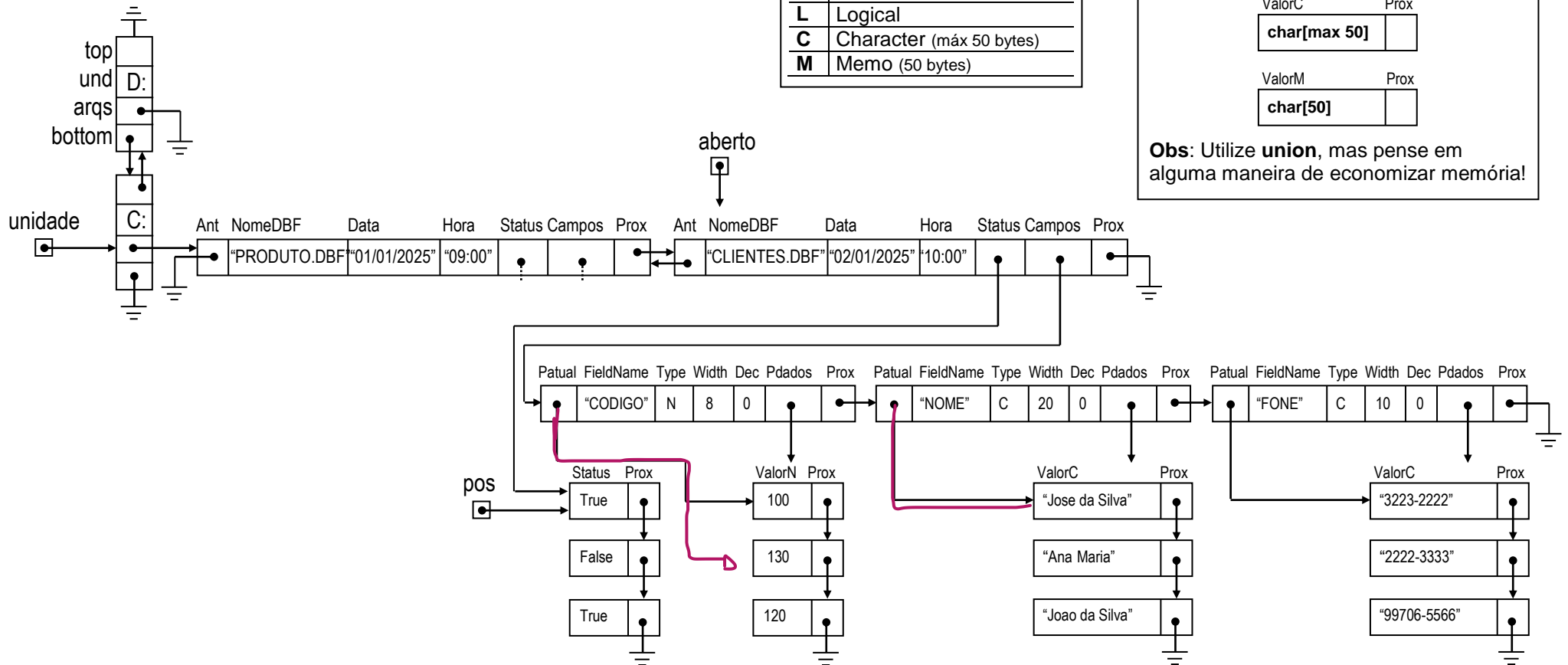
Estruturas de Dados II – Trabalho 1º Bimestre – 2025

Professores: Francisco Assis da Silva

dBase com Listas Encadeadas Dinâmicas

Trabalho em dupla

Considere as seguintes estruturas dinâmicas:



Comandos do dBase que devem ser implementados no trabalho:

Tratei o CREATE e SET DEFAULT TO

1) **SET DEFAULT TO:** no caso do nosso trabalho apenas para o D: e C: Precisa tratar a string

Exemplos:

SET DEFAULT TO D:

SET DEFAULT TO C:

2) **CREATE :** Cria um novo arquivo .DBF (em memória segundo as estruturas definidas anteriormente). Precisa tratar a string

Exemplo:

CREATE CLIENTES.DBF

Falta criar os campos de dentro do arquivo (ARRUMAR)

	Field Name	Type	Width	Dec
1	CODIGO	Numeric	8	0
2	NOME	Character	20	
3	FONE	Character	10	


```
CREATE ||C: > CLIENTES
Enter the field name and type for each field.
Character fields are 1 to 255 characters long.
```

3) **DIR:** para mostrar todos os arquivos .DBF criados na unidade atual. Apenas strcmp para comparar

Exemplo:

DIR

```
. DIR
Database Files      # Records    Last Update    Size
CLIENTES.DBF       0             02/01/25       130

130 bytes in      1 files.
1023932928 bytes remaining on drive.
```



```
Command Line ||C: > CLIENTES ||Rec: None
Enter a dBASE III PLUS command.
```

4) **QUIT:** Encerra o ambiente do dBase.

5) **USE:** Abre um arquivo .DBF. Precisa tratar a string

Exemplo:

```
USE CLIENTES.DBF
```

6) **LIST STRUCTURE:** Lista a estrutura criada para o arquivo .DBF.

Ainda tenho que arrumar para apenas exibir

Exemplo:

```
LIST STRUCTURE
```

```
. LIST STRUCTURE
Structure for database: C:\CLIENTES.DBF
Number of data records:      0
Date of last update   : 02/01/25
Field  Field Name  Type      Width  Dec
   1  CODIGO      Numeric     8
   2  NOME        Character    20
   3  FONE        Character    10
** Total **
      39

Command Line  ||<C:>||CLIENTES          ||Rec: N
Enter a dBASE III PLUS comma
```

7) **APPEND:** Insere novo registro ao final.

Apenas strcmp para comparar

Exemplo:

```
APPEND
```

```
CODIGO      100
NOME        Jose da Silva
FONE        3223-2222

APPEND      ||<C:>||CLIENTES
```

8) **LIST**: Lista todos os campos.

Apenas strcmp para comparar

Exemplos:

LIST

```
. LIST
Record#    CODIGO NOME                FONE
1          100 Jose da Silva          3223-2222
2          130 Ana Maria              2222-3333
3          120 Joao da Silva          99706-5566

Command Line ||<C:>||CLIENTES ||Rec: EOF/3

Enter a dBASE III PLUS command.
```

LIST FOR NOME = "Jo"

```
. LIST FOR NOME = "Jo"
Record#    CODIGO NOME                FONE
1          100 Jose da Silva          3223-2222
3          120 Joao da Silva          99706-5566

Command Line ||<C:>||CLIENTES ||Rec: EOF/3

Enter a dBASE III PLUS command.
```

9) **CLEAR**: Limpa a tela do ambiente do dBase.

Apenas strcmp para comparar

10) **LOCATE**: Localiza registro(s), informando o campo e o conteúdo a ser buscado.

Precisa tratar a string

Exemplo:

LOCATE FOR NOME = "Joao da Silva"

```
. LOCATE FOR NOME = "Joao da Silva"
Record =      3

Command Line ||<C:>||CLIENTES ||Rec: 3/3

Enter a dBASE III PLUS command.
```

11) **GOTO**: Vai para determinado registro.

Apenas strcmp para comparar

Exemplo:

GOTO 2

```
. GOTO 2

Command Line ||<C:>||CLIENTES ||Rec: 2/3

Enter a dBASE III PLUS command.
```

12) **DISPLAY:** Mostra o registro atual.

Apenas strcmp para comparar

Exemplo:

DISPLAY

```
. DISPLAY
Record#  CODIGO NOME          FONE
      2    130 Ana Maria      2222-3333
Command Line  ||C:>>||CLIENTES          ||Rec: 2/3
Enter a dBASE III PLUS command
```

13) **EDIT:** Permite alterar os dados do registro atual.

Apenas strcmp para comparar

Exemplo:

EDIT

```
CODIGO  130
NOME    Ana Maria
FONE    2222-3333
EDIT    ||C:>>||CLIENTES          ||Rec: 2/3
```

14) **DELETE:** Marca o registro atual para exclusão (um tipo de exclusão lógica).

Exemplo:

DELETE

Apenas strcmp para comparar

```
. DELETE
      1 record deleted
Command Line  ||C:>>||CLIENTES          ||Rec: 2/3
Enter a dBASE III PLUS command
```

DELETE ALL {este comando marca todos os registros para exclusão}

15) **RECALL**: Reativa o(s) registro(s) marcado(s) para exclusão.

Exemplo:

RECALL

[Apenas strcmp para comparar](#)

```
. RECALL
    1 record recalled
.
Command Line  ||C:>||CLIENTES          ||Rec: 2/3
Enter a dBASE III PLUS command
```

RECALL ALL {Reativa todos os registros marcados para exclusão, desde que o comando SET DELETED OFF seja executado previamente}

16) **SET DELETED**

[Apenas strcmp para comparar](#)

Exemplo:

SET DELETED OFF {Este comando permite que se possa visualizar e ou reativar registros marcados para exclusão}

SET DELETED ON {Este comando esconde os registros marcados para exclusão}

17) **PACK**: Exclui fisicamente os registros marcados para exclusão. [Apenas strcmp para comparar](#)

18) **ZAP**: Este comando exclui todos os registros do arquivo .DBF em uso. [Apenas strcmp para comparar](#)

O trabalho deverá possuir interfaces semelhantes ao dBase original (década de 80, 90). Na tela principal do programa, os comandos deverão ser escritos e, portanto, interpretados e validados. Para isso, deverá ser feita uma verificação para saber se a sintaxe do comando foi escrita corretamente e se os parâmetros estão corretos.

Alguns outros comandos do dBase podem ser implementados:

1) **MODIFY STRUCTURE**: Permite modificar a estrutura do arquivo .DBF (alterar os campos existentes ou criar novos).

2) **SORT**: Ordena os dados em relação a algum campo especificado.