

# Atividade Prática: Busca Sequencial e por RRN

Organização e Recuperação de Dados  
Profa. Valéria

UEM – CTC – DIN

# Programa 1

- Implemente o pseudocódigo *busca\_seq*
- O programa *busca\_seq* faz uma busca sequencial em um arquivo de registros de tamanho variável no formato gravado pelo programa *escreve\_registros*
  - O programa *escreve\_registros* foi criado como parte da Atividade prática 1 – Campos e Registros

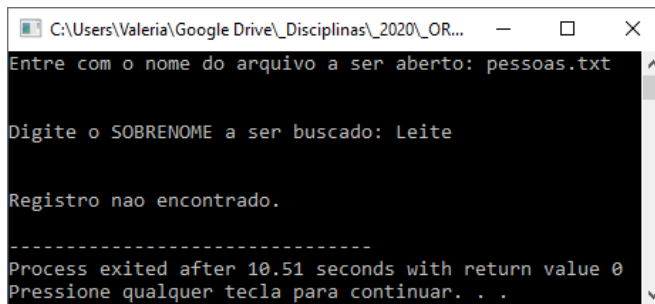
Exemplo de registros com **indicador de tamanho no início**

**42**Silva|Alan|Rua Tiete 123|Maringa|PR|87100|**38**Soares|Andre|Rua Braga 34|Sarandi|...

2 bytes                      42 bytes

# Programa 1

- Inicialmente, o programa deve solicitar o nome do arquivo a ser pesquisado
  - Caso o arquivo não exista, o programa deve imprimir uma mensagem de erro e terminar
- Após a abertura do arquivo, o programa deve solicitar um SOBRENOME, o qual será utilizado como chave primaria de busca
- Se o registro correspondente for encontrado, o mesmo deve ser apresentado em tela, finalizando a execução
  - Caso contrário, o programa deve imprimir “Registro não encontrado” e terminar

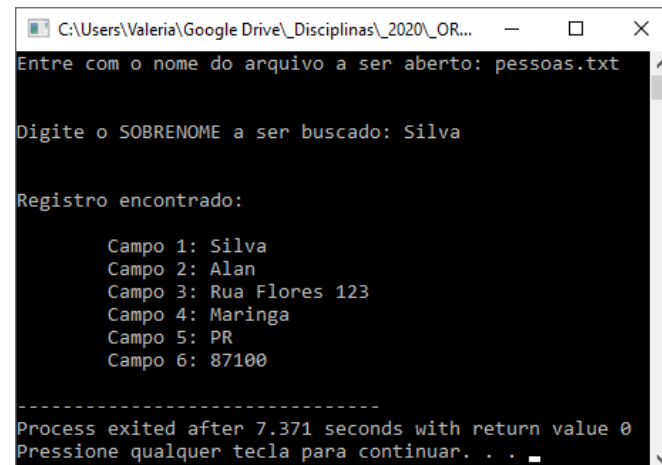


```
C:\Users\Valeria\Google Drive\Disciplinas\2020\OR... — □ ×
Entre com o nome do arquivo a ser aberto: pessoas.txt

Digite o SOBRENOME a ser buscado: Leite

Registro nao encontrado.

-----
Process exited after 10.51 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```



```
C:\Users\Valeria\Google Drive\Disciplinas\2020\OR... — □ ×
Entre com o nome do arquivo a ser aberto: pessoas.txt

Digite o SOBRENOME a ser buscado: Silva

Registro encontrado:

    Campo 1: Silva
    Campo 2: Alan
    Campo 3: Rua Flores 123
    Campo 4: Maringa
    Campo 5: PR
    Campo 6: 87100

-----
Process exited after 7.371 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

Busca sequencial  
em um arquivo com  
registros de tamanho  
variável no formato  
gravado pelo  
programa  
**escreve\_registros.py**

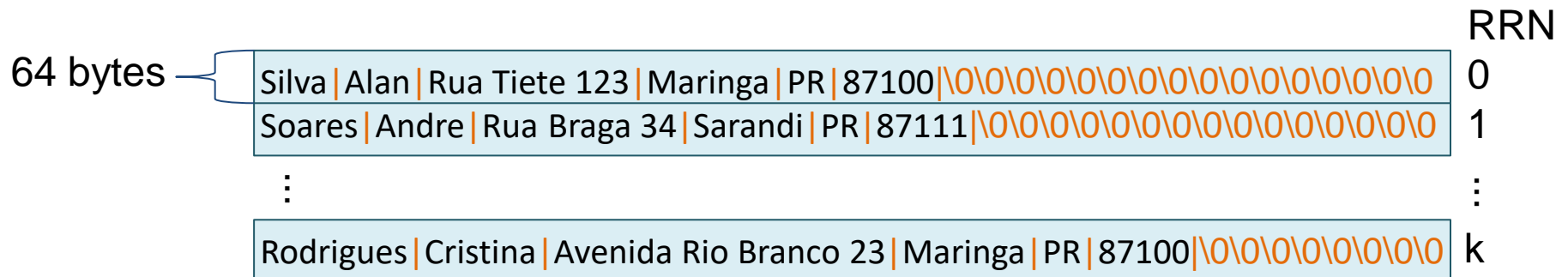
A função **leia\_reg()**  
mencionada neste  
pseudocódigo é a  
mesma que foi utilizada  
no programa  
**le\_registros.py**

```
PROGRAMA: busca_seq

  receba o nome do arquivo a ser aberto em NOME_ARQ
  abra NOME_ARQ para leitura binária e chame-o ENTRADA
  Se NOME_ARQ não existir, termine o programa
  receba o sobrenome a ser buscado em CHAVE
  inicialize ACHOU com falso
  chame a função leia_reg(ENTRADA) e armazene o retorno em REG
enquanto REG for diferente de vazio e não ACHOU faça
    faça SOBRENOME receber o 1º campo de REG
    # sobrenome = reg.split(sep='|')[0]
    se SOBRENOME é igual a CHAVE então
      ACHOU recebe verdadeiro
    senão chame leia_reg(ENTRADA) e armazene em REG
se ACHOU então
  # Lembre-se que reg.split(sep='|') retorna
  uma lista com os campos contidos em REG
  para cada CAMPO em REG faça
    escreva CAMPO na tela
  senão imprima uma mensagem de erro
  feche ENTRADA
fim PROGRAMA
```

# Programa 2

- Implemente o pseudocódigo *busca\_rrn*
- O programa *busca\_RRN* faz uma busca direta em um arquivo de **registros de tamanho fixo**
  - Usaremos registros de 64 bytes
    - Registros de tamanho fixo com campos de tamanho variável
    - As sobras de espaço dentro de um registro são ser preenchidas com o caractere '\0'
  - O arquivo inicia com um **cabeçalho de 4 bytes** que armazena o **total de registros** do arquivo



Busca direta  
em um arquivo de  
registros de tamanho  
fixo no formato  
explicado no slide  
anterior

O número **total de registros** no arquivo  
está gravado no  
cabeçalho como um  
número de 4 bytes

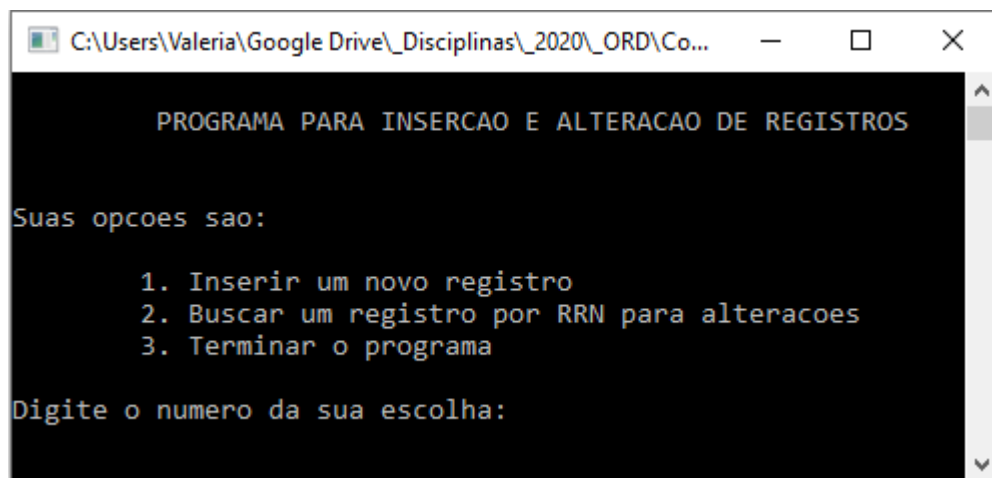
```
PROGRAMA: busca_rrn

  receba o nome do arquivo a ser aberto em NOME_ARQ
  abra NOME_ARQ para leitura binária e chame-o ENTRADA
  Se NOME_ARQ não existir, termine o programa
  leia 4 bytes de ENTRADA e armazene de CAB
  Converta CAB para inteiro armazene em TOTAL_REG
    # utilize o método int.from_bytes() para converter CAB
  receba o RRN do registro a ser lido
  se RRN >= TOTAL_REG então
    imprima uma msg de erro e termine
  calcule o OFFSET de leitura /* RRN * 64 + 4 */
  faça seek para OFFSET
  leia o registro e decodifique-o em REG
    # utilize o método decode() para decodificar REG
    # Lembre-se que reg.split(sep='|') retorna
      uma lista com os campos contidos em REG
  para cada CAMPO em REG faça
    escreva CAMPO na tela
  feche ENTRADA
fim PROGRAMA
```

# Programa 3

- **Implemente o pseudocódigo *busca\_e\_atualiza***
- Diferentemente dos programas feitos até agora, o programa *busca\_e\_atualiza* permite que:
  - Se abra um arquivo existente (ou se crie, caso não exista)
  - Se insira novos registros (sempre no final do arquivo)
  - Se **busque um registro pelo seu RRN para alterações** (busca e possível regravação do registro)
- Para permitir a busca por RRN, o programa deve utilizar **registros de tamanho fixo**
  - Adotaremos o mesmo formato de **registros de tamanho fixo com campos de tamanho variável utilizado pelo Programa 2**

# Programa 3



```
C:\Users\Valeria\Google Drive\Disciplinas\2020\ORD\Co...  
  
PROGRAMA PARA INSERCAO E ALTERACAO DE REGISTROS  
  
Suas opcoes sao:  
  
1. Inserir um novo registro  
2. Buscar um registro por RRN para alteracoes  
3. Terminar o programa  
  
Digite o numero da sua escolha:
```



# Programa 3

```
C:\Users\Valeria\Google Drive\_Disciplinas\_2020\_ORD\Co...  
  
PROGRAMA PARA INSERCAO E ALTERACAO DE REGISTROS  
  
Suas opcoes sao:  
  
1. Inserir um novo registro  
2. Buscar um registro por RRN para alteracoes  
3. Terminar o programa  
  
Digite o numero da sua escolha:
```

```
C:\Users\Valeria\Google Drive\_Disciplinas\_2020\_OR...  
Digite o numero da sua escolha: 1  
Digite os dados para o novo registro  
  
Sobrenome: Souza  
Primeiro nome: Ricardo  
Endereco: Rua Paicandu, 234  
Cidade: Maringa  
Estado: PR  
CEP: 87100
```

# Programa 3

```
C:\Users\Valeria\Google Drive\_Disciplinas\_2020\_ORD\Co...  
  
PROGRAMA PARA INSERCAO E ALTERACAO DE REGISTROS  
  
Suas opcoes sao:  
  
1. Inserir um novo registro  
2. Buscar um registro por RRN para alteracoes  
3. Terminar o programa  
  
Digite o numero da sua escolha:
```

```
C:\Users\Valeria\Google Drive\_Disciplinas\_2020\_ORD\Codigo C...  
  
Digite o numero da sua escolha: 2  
  
Digite o RRN do registro: 0  
  
Conteudo do registro  
Silva  
Joao  
Rua Braga 987  
Floresta  
PR  
87200  
  
Voce quer modificar este registro?  
Responda S or N, seguido de <ENTER> ==>
```

# Programa 3

```
C:\Users\Valeria\Google Drive\_Disciplinas\_2020\_ORD\Codigo C...  
Digite o numero da sua escolha: 2  
  
Digite o RRN do registro: 0  
  
Conteudo do registro  
    Silva  
    Joao  
    Rua Braga 987  
    Floresta  
    PR  
    87200  
  
Voce quer modificar este registro?  
    Responda S or N, seguido de <ENTER> ==>
```

```
C:\Users\Valeria\Google Drive\_Disciplinas\_2020\_ORD\Codigo C...  
Voce quer modificar este registro?  
    Responda S or N, seguido de <ENTER> ==>S  
  
Digite os novos dados do registro:  
  
    Sobrenome: _
```

## Abertura ou criação um arquivo de registros de tamanho fixo (64b)

Os registros podem  
ser inseridos  
ou buscados

## Registros são buscados por RRN

Armazene o número  
**total de registros** no  
cabeçalho do arquivo  
usando 4 bytes

O cabeçalho deve  
ser gravado na  
criação do arquivo e  
atualizado ao final de  
cada execução

PROGRAMA: **busca\_e\_atualiza**

receba o nome do arquivo em NOME\_ARQ

abra o arquivo NOME\_ARQ para L/E e chame-o de ARQ # *'r+b'*

**se** o arquivo NOME\_ARQ não existir **então**

crie e abra NOME\_ARQ para L/E com o nome lógico ARQ # *'w+b'*

faça TOTAL\_REG = 0 e escreva-o em ARQ com 4 bytes

**senão** leia o cabeçalho de ARQ e armazene-o em TOTAL\_REG

receba a escolha do usuário em OPCA0 # *(1)inserir (2)buscar (3)sair*

**enquanto** (OPCA0 < 3) **faça**

**se** OPÇÃO == 1:

receba todos os campos do registro e concatene em REG

codifique e preencha REG até 64b

*# use encode() para codificar e ljust(64, b'\0') para preencher*

calcule o OFFSET de gravação # *TOTAL\_REG \* 64 + 4*

faça seek para OFFSET e escreva REG em ARQ

incremente TOTAL\_REG

**senão se** OPCA0 == 2:

receba o RRN

**se** RRN >= TOTAL\_REG imprima msg de erro

**senão** calcule o OFFSET de leitura # *RRN \* 64 + 4*

faça seek para OFFSET

leia o registro REG e mostre-o na tela

receba a opção sobre uma possível alteração em ALTERAR

**se** ALTERAR **então**

receba todos os campos do registro e concatene em REG

codifique e preencha REG até 64b

faça seek para OFFSET e escreva REG em ARQ

receba a escolha do usuário em OPCA0

*# fim enquanto -- CONTINUAÇÃO NO PRÓXIMO SLIDE*

Abertura ou criação  
um arquivo de  
**registros de  
tamanho fixo (64b)**

Os registros podem  
ser inseridos  
ou buscados

Registros são  
**buscados  
por RRN**

Armazene o número  
**total de registros** no  
**cabeçalho** do arquivo  
usando 4 bytes

O cabeçalho deve  
ser gravado na  
criação do arquivo e  
atualizado ao final de  
cada execução

```
# -- CONTINUAÇÃO busca_e_atualiza  
faça seek para o início de ARQ  
escreva TOTAL_REG em ARQ com 4 bytes  
feche ARQ  
fim PROGRAMA
```