

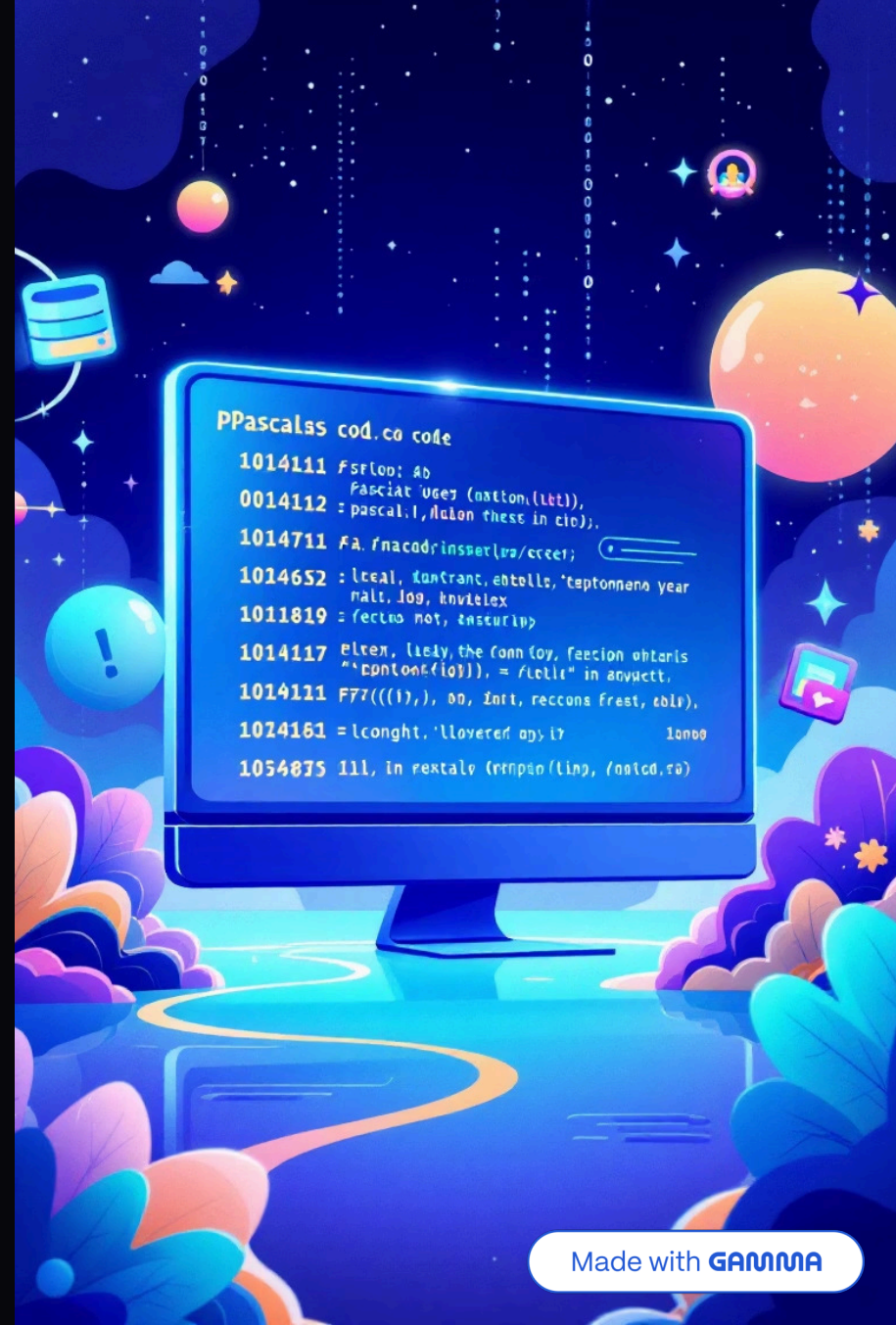
Aula 09 | Manipulação de Arquivos

Até agora, vimos como manipular variáveis simples, vetores, records e outros tipos de dados. Porém, todas essas informações só existem **enquanto o programa está em execução**. Se quisermos armazenar dados de forma **permanente**, precisamos trabalhar com **arquivos**.

Em Pascal, a manipulação de arquivos é bastante poderosa, permitindo desde a gravação e leitura de **textos simples** até o gerenciamento de **estruturas complexas com registros**.

Nesta aula, aprenderemos a trabalhar com:

- Arquivos de texto (TextFile);
- Arquivos binários (File of ...);
- Arquivos de registros (Record);
- Operações de localizar, adicionar e excluir registros em arquivos.



Arquivos de Texto

Um arquivo de texto é composto por caracteres legíveis, como .txt. No Pascal, trabalhamos com arquivos de texto usando o tipo **TextFile**.

1

Procedimentos e Funções

- **Assign**(arquivo, 'nome.txt')
- **Rewrite**(arquivo)
- **Append**(arquivo)
- **Reset**(arquivo)
- **Close**(arquivo)
- **Read / ReadLn**
- **Write / WriteLn**

2

Exemplo: Gravando

```
program GravarTexto;
var
  arq: TextFile;
  i: integer;
begin
  Assign(arq, 'numeros.txt');
  Rewrite(arq);
  for i := 1 to 10 do
    Writeln(arq, 'Número: ', i);
  Close(arq);
end.
```

3

Exemplo: Lendo

```
program LerTexto;
var
  arq: TextFile;
  linha: string;
begin
  Assign(arq, 'numeros.txt');
  Reset(arq);
  while not Eof(arq) do
    begin
      Readln(arq, linha);
      Writeln('Linha lida: ', linha);
    end;
  Close(arq);
end.
```

Estes exemplos demonstram a simplicidade de ler e escrever dados textuais. Lembre-se sempre de fechar o arquivo após as operações para evitar perda de dados ou corrupção.

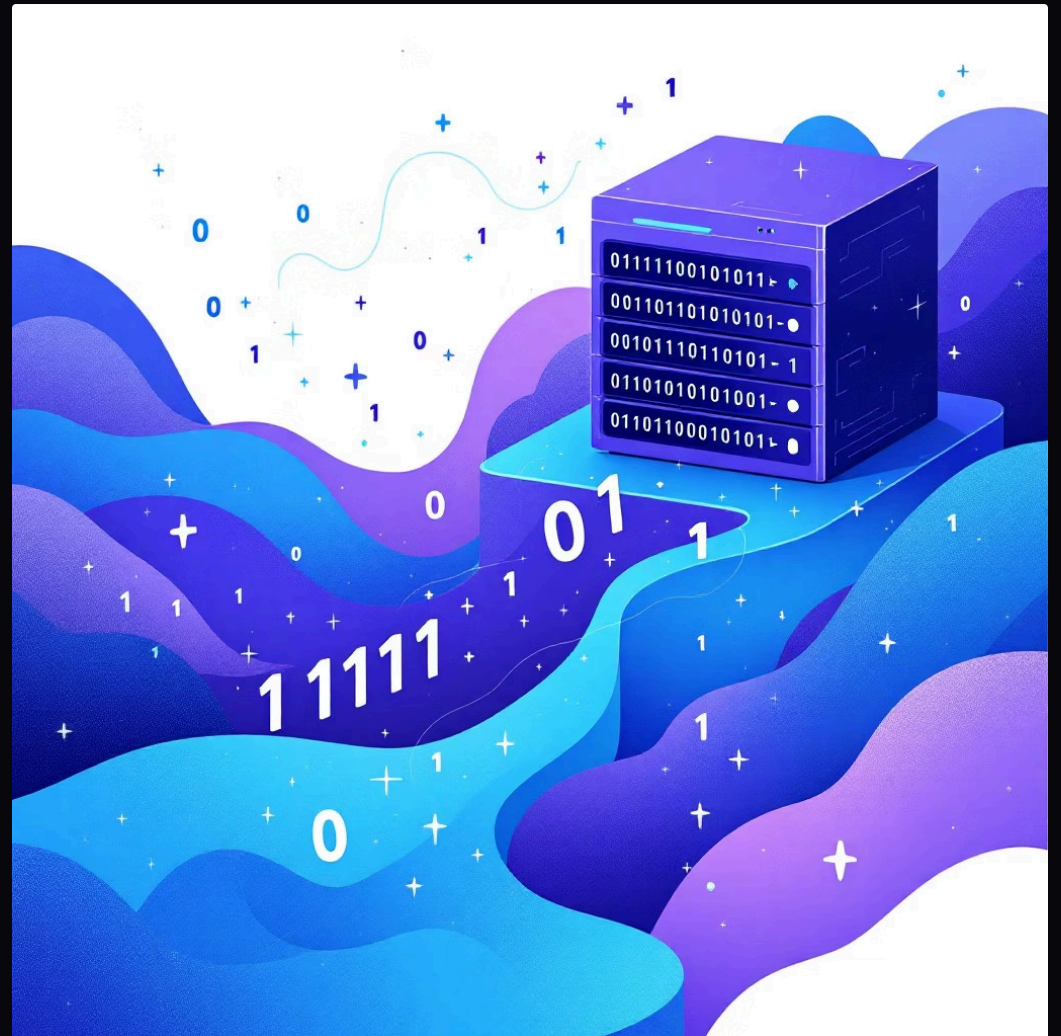
Arquivos Binários (File of)

Enquanto os arquivos de texto armazenam caracteres legíveis, os arquivos binários guardam os dados de forma compacta e mais eficiente, ideal para dados estruturados.

A declaração é feita com **File of Tipo**, indicando o tipo de dado que será armazenado.

Gravando Inteiros

```
program ArquivoBinario;
var
  arq: file of integer;
  i: integer;
begin
  Assign(arq, 'dados.bin');
  Rewrite(arq);
  for i := 1 to 5 do
    Write(arq, i * 10);
  Close(arq);
end.
```



Lendo Inteiros

```
program LerBinario;
var
  arq: file of integer;
  valor: integer;
begin
  Assign(arq, 'dados.bin');
  Reset(arq);
  while not Eof(arq) do
  begin
    Read(arq, valor);
    Writeln('Valor: ', valor);
  end;
  Close(arq);
end.
```



Arquivos binários são excelentes para armazenar grandes volumes de dados de um mesmo tipo, otimizando espaço e velocidade de acesso.

Arquivos de Registros

Um **registro (record)** pode ser armazenado diretamente em um arquivo binário, o que é extremamente útil para sistemas de cadastro complexos.

Exemplo: Estrutura de Cliente

```
type
  TCliente = record
    cpf: string[11];
    nome: string[50];
    telefone: string[15];
  end;

var
  arqClientes: file of TCliente;
```

Gravando Registros

```
program GravarClientes;
type
  TCliente = record
    cpf: string[11];
    nome: string[50];
    telefone: string[15];
  end;
var
  arq: file of TCliente;
  cliente: TCliente;
begin
  Assign(arq, 'clientes.dat');
  Rewrite(arq);
  cliente.cpf := '12345678901';
  cliente.nome := 'Joao Silva';
  cliente.telefone := '11999999999';
  Write(arq, cliente);
  cliente.cpf := '98765432100';
  cliente.nome := 'Maria Oliveira';
  cliente.telefone := '11888888888';
  Write(arq, cliente);
  Close(arq);
end.
```

Lendo Registros

```
program LerClientes;
type
  TCliente = record
    cpf: string[11];
    nome: string[50];
    telefone: string[15];
  end;
var
  arq: file of TCliente;
  cliente: TCliente;
begin
  Assign(arq, 'clientes.dat');
  Reset(arq);
  while not Eof(arq) do
    begin
      Read(arq, cliente);
      Writeln('CPF: ', cliente.cpf, ' Nome: ', cliente.nome, '
Telefone: ', cliente.telefone);
    end;
  Close(arq);
end.
```


Localizando Registros em um Arquivo

Para encontrar um registro específico em um arquivo binário, percorremos o arquivo registro por registro até que o critério de busca seja satisfeito.



```
program BuscarCliente;
type
  TCliente = record
    cpf: string[11];
    nome: string[50];
    telefone: string[15];
  end;
var
  arq: file of TCliente;
  cliente: TCliente;
  buscado: string[11];
  encontrado: boolean;
begin
  Assign(arq, 'clientes.dat');
  Reset(arq);
  Write('Digite o CPF para buscar: ');
  Readln(buscado);
  encontrado := false;
  while not Eof(arq) do
  begin
    Read(arq, cliente);
    if cliente.cpf = buscado then
    begin
      Writeln('Cliente encontrado: ', cliente.nome, ' - ',
cliente.telefone);
      encontrado := true;
      break;
    end;
  end;
  if not encontrado then
    Writeln('Cliente não encontrado.');
```

```
Close(arq);
end.
```

A função Eof (End of File) é fundamental para controlar o laço de leitura, garantindo que a busca pare ao final do arquivo.

Adicionando Registros em um Arquivo

A adição de novos registros em um arquivo binário existente requer a abertura do arquivo em modo de escrita, mas com o ponteiro posicionado no final.



1. Abrir Arquivo

Abra o arquivo em modo de leitura/escrita usando **Reset(arq)**.



2. Posicionar Ponteiro

Use **Seek(arq, FileSize(arq))** para mover o ponteiro para a última posição do arquivo.



3. Escrever Novo Registro

Utilize **Write(arq, novoRegistro)** para adicionar o novo registro ao final do arquivo.

Este método evita a sobrescrita de dados existentes, garantindo que o arquivo seja atualizado corretamente com os novos dados. Após a operação, não se esqueça de **Close(arq)**.

Excluindo Registros em um Arquivo

Em Pascal, a exclusão de registros em arquivos binários não é direta. A estratégia padrão envolve a recriação do arquivo sem o registro indesejado.

01

Criar Arquivo Temporário

Crie um novo arquivo vazio para armazenar os registros que serão mantidos.

02

Copiar Registros Válidos

Leia o arquivo original registro por registro. Copie para o arquivo temporário **todos** os registros, exceto aquele que você deseja excluir.

03

Excluir Original e Renomear

Após a cópia, exclua o arquivo original com **Erase(arq)** e renomeie o arquivo temporário para o nome do arquivo original com **Rename(temp, 'nome_original.dat')**.

Este processo, embora indireto, é o método mais seguro para manter a integridade dos dados durante a exclusão. A precisão na lógica de cópia é crucial para evitar perdas de dados.

Exercícios

Para consolidar seu aprendizado, pratique com os exercícios abaixo. Eles cobrem desde operações básicas até a manipulação avançada de registros.

- Crie um programa que grave os nomes de 5 pessoas em um arquivo de texto e depois leia-os na tela.
- Crie um programa que leia um arquivo de texto linha por linha e conte quantas linhas ele possui.
- Grave 20 números inteiros em um arquivo binário e depois leia-os de volta mostrando na tela.
- Crie um programa que leia um arquivo binário de inteiros e calcule a soma de todos os valores.
- Defina um record para armazenar livros (título, autor, ano) e grave 3 registros em um arquivo.
- Faça um programa que leia os registros do arquivo de livros e exiba apenas os livros publicados após 2000.
- Implemente a busca de clientes em um arquivo binário pelo CPF (como no exemplo).
- Crie uma rotina para adicionar novos clientes a um arquivo já existente.
- Implemente a exclusão de clientes pelo CPF, recriando o arquivo sem o registro excluído.
- Crie um sistema simples de cadastro de alunos usando arquivos de registros, com opções de adicionar, listar, buscar e excluir.

A resolução desses exercícios reforçará os conceitos apresentados e preparará você para desafios mais complexos na programação Pascal.



Com isso, você tem uma base sólida para trabalhar com arquivos em Pascal.

Desde operações simples em **texto** até sistemas de cadastro completos com **registros**.

Continue praticando para dominar completamente esta ferramenta poderosa!