

A large, solid orange oval shape that serves as the background for the text.

**ARCHIVOS**

# Concepto

**Un archivo es un tipo de dato estructurado, dinámico con la particularidad de que se almacena en memoria Secundaria**

**Nota:**

**Los datos no se pierden al salir del programa**

# **Tipos de Acceso**

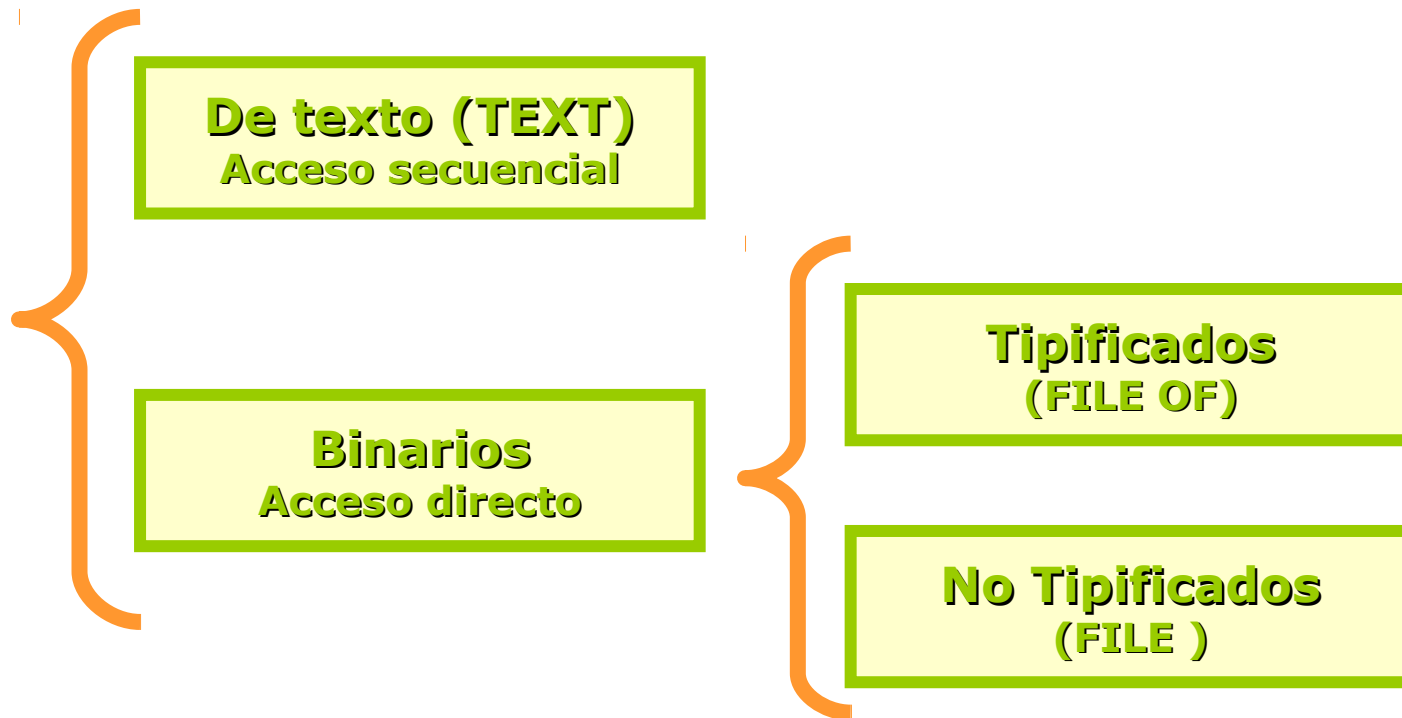


A diagram illustrating types of access. At the top, a yellow oval contains the title 'Tipos de Acceso'. Below it, a large orange curly brace groups two yellow rectangular boxes. The top box is labeled 'SECUENCIAL' and the bottom box is labeled 'DIRECTO'.

**SECUENCIAL**

**DIRECTO**

# Clasificación (para Pascal)



# **Extensión de Archivos**

**(creados en Pascal)**

**Regla ( SEGÚN manual del Turbo Pascal )**

**“Los archivos de texto estandar de Turbo Pascal tienen la extensión .TXT, por consiguiente nombre A sus archivos de datos (creados por usted) con la Extensión .DAT”**

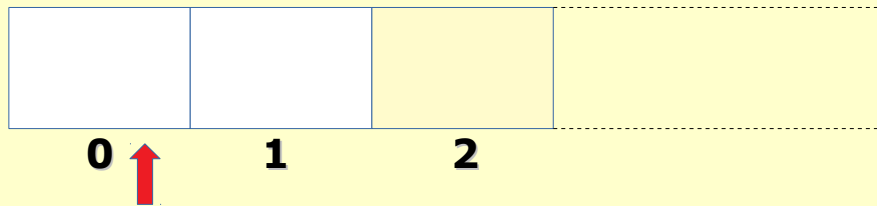
# **Archivos Random o Aleatorios**

## **Concepto**

**Se estructuran de acuerdo a un tipo definido  
previamente**

**Su acceso es directo (random), con la posición del  
elemento. También se puede acceder secuencialmente**

# Archivos Aleatorios Gráficamente



PUNTERO INTERNO DEL ARCHIVO

**Declaración**

**TYPE**

**T\_DATO = ...;**

**T\_ARCHIVO = FILE OF T\_DATO;**

**VAR**

**ARCH:T\_ARCHIVO**

# Operaciones

**Asignar (Assign (Arch, Ruta))**

**Crear (Rewrite (Arch))**

**Abrir (Reset (Arch))**

**Cerrar (Close (Arch))**

**Renombrar(Rename)**

**Borrar (Erase)**



# **Ejemplo Archivo de Reales en Pascal**

**VER EJEMPLO**

•  
•  
•

**ARCHIVO\_REAL //LAZARUS**

# **Archivos Tipo Texto**

## **Concepto**

**Se estructuran línea a línea  
Son de acceso secuencial  
No se puede implementar el acceso directo pues las  
líneas no siempre tienen la misma longitud**

# Ejemplo

## Archivo de texto en Pascal

```
program ARCHIVO_TEXTO_2020;  
• USES CRT;  
• CONST  
•   RUTA='C:\Archivos_texto\ARCHIVO_2020.txt';  
•  
• VAR  
•   CAD:STRING;  
•   TE:TEXT;  
•  
• begin  
•   ASSIGN (TE, RUTA);  
•   RESET (TE);  
•   WHILE (NOT EOF(TE)) DO  
•     BEGIN  
•       READLN (TE,CAD);  
•       WRITELN (CAD);  
•     end;  
•   CLOSE (TE);  
•   READKEY;  
• end.
```