

Informática I  
Manejo de archivos

Claudio J. Paz

9 de noviembre de 2017



## Archivos

- Concepto de *stream*

## Archivos

- Concepto de *stream*
  - stdin

## Archivos

- Concepto de *stream*
  - stdin
  - stdout

## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre



# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof

## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc

## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets

## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread
  - fscanf



## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread
  - fscanf
- Funciones de escritura

## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread
  - fscanf
- Funciones de escritura
  - fputc

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread
  - fscanf
- Funciones de escritura
  - fputc
  - fputs

# Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread
  - fscanf
- Funciones de escritura
  - fputc
  - fputs
  - fwrite

## Archivos

- Concepto de *stream*
  - stdin
  - stdout
  - stderr
  - archivos
- Funciones de apertura y cierre
  - fopen
  - fclose
  - feof
- Funciones de lectura
  - fgetc
  - fgets
  - fread
  - fscanf
- Funciones de escritura
  - fputc
  - fputs
  - fwrite
  - fprintf



# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```



# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

```
0
1
2
3
4
5
6
7
8
9
```

- fprintf está definida en stdio.h

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

```
0
1
2
3
4
5
6
7
8
9
```

- `fprintf` está definida en `stdio.h`
- `stdin`, `stdout` y `stderr` pueden ser considerados *archivos*

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

```
0
1
2
3
4
5
6
7
8
9
```

- `fprintf` está definida en `stdio.h`
- `stdin`, `stdout` y `stderr` pueden ser considerados *archivos*
- `stdout` *apunta* a la salida por pantalla

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

```
0
1
2
3
4
5
6
7
8
9
```

- `fprintf` está definida en `stdio.h`
- `stdin`, `stdout` y `stderr` pueden ser considerados *archivos*
- `stdout` *apunta* a la salida por pantalla
- equivalente a `printf("%d", a[i]);`



# Archivos

```
#include<stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt","w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

# Archivos

```
#include<stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```



# Archivos

```
#include<stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
```

```
> cat salida.txt
```

```
0
1
2
3
4
5
6
7
8
9
```

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`
- Los argumentos son el nombre del archivo y el tipo de acceso

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`
- Los argumentos son el nombre del archivo y el tipo de acceso
  - `w` para escribir (write)

# Archivos

```
#include<stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`
- Los argumentos son el nombre del archivo y el tipo de acceso
  - `w` para escribir (write)
  - `r` para leer (read)

# Archivos

```
#include<stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`
- Los argumentos son el nombre del archivo y el tipo de acceso
  - `w` para escribir (write)
  - `r` para leer (read)
  - `a` para agregar (append)

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`
- Los argumentos son el nombre del archivo y el tipo de acceso
  - `w` para escribir (write)
  - `r` para leer (read)
  - `a` para agregar (append)
- Devuelve un `FILE *` que apunta al archivo

# Archivos

```
#include <stdio.h>

int main(void)
{
    int i;
    int a[] = {0,1,2,3,4,5,6,7,8,9};

    fclose(stdout);
    stdout = fopen("salida.txt", "w");

    for ( i = 0 ; i < 10 ; i++ )
        fprintf(stdout, "%d\n", a[i]);

    return 0;
}
> cat salida.txt
0
1
2
3
4
5
6
7
8
9
```

- `fopen` está definida en `stdio.h`
- Los argumentos son el nombre del archivo y el tipo de acceso
  - `w` para escribir (write)
  - `r` para leer (read)
  - `a` para agregar (append)
- Devuelve un `FILE *` que apunta al archivo
- `fclose` cierra el archivo o *stream* apuntado por el argumento





# Archivos

```
#include<stdio.h>

int main(void)
{
    char fruta[80];

    fscanf(stdin, "%s", fruta);

    printf("ingresó: %s\n", fruta);

    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    char fruta[80];

    fscanf(stdin, "%s", fruta);

    printf("ingresó: %s\n", fruta);

    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    char fruta[80];

    fscanf(stdin, "%s", fruta);

    printf("ingresó: %s\n", fruta);

    return 0;
}
```

- `fscanf` está definida en `stdio.h`

# Archivos

```
#include<stdio.h>

int main(void)
{
    char fruta[80];

    fscanf(stdin, "%s", fruta);

    printf("ingresó: %s\n", fruta);

    return 0;
}
```

- fscanf está definida en stdio.h

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
> ./imprime-fruta
manzana
ingresó: manzana
```

# Archivos

```
#include<stdio.h>

int main(void)
{
    char fruta[80];

    fscanf(stdin, "%s", fruta);

    printf("ingresó: %s\n", fruta);

    return 0;
}
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
> ./imprime-fruta
manzana
ingresó: manzana
```

# Archivos

```
#include<stdio.h>

int main(void)
{
    char fruta[80];
    fscanf(stdin, "%s", fruta);
    printf("ingresó: %s\n", fruta);
    return 0;
}
```

```
#include<stdio.h>

int main(void)
{
    printf("%s", "banana\n");
    return 0;
}
```

# Archivos

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    char fruta[80];
```

```
    fscanf(stdin, "%s", fruta);
```

```
    printf("ingresó: %s\n", fruta);
```

```
    return 0;
```

```
}
```

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("%s", "banana\n");
```

```
    return 0;
```

```
}
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
```



# Archivos

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    char fruta[80];
```

```
    fscanf(stdin, "%s", fruta);
```

```
    printf("ingresó: %s\n", fruta);
```

```
    return 0;
```

```
}
```

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("%s", "banana\n");
```

```
    return 0;
```

```
}
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00t.c -o imprime-banana
```

# Archivos

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    char fruta[80];
```

```
    fscanf(stdin, "%s", fruta);
```

```
    printf("ingresó: %s\n", fruta);
```

```
    return 0;
```

```
}
```

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("%s", "banana\n");
```

```
    return 0;
```

```
}
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
> gcc -Wall -std=c90 -pedantic-errors archivos-00t.c -o imprime-banana
> ./imprime-banana
banana
```

# Archivos

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    char fruta[80];
```

```
    fscanf(stdin, "%s", fruta);
```

```
    printf("ingresó: %s\n", fruta);
```

```
    return 0;
```

```
}
```

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("%s", "banana\n");
```

```
    return 0;
```

```
}
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
> gcc -Wall -std=c90 -pedantic-errors archivos-00t.c -o imprime-banana
> ./imprime-banana
banana
> ./imprime-banana | ./imprime-fruta
```

# Archivos

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    char fruta[80];
```

```
    fscanf(stdin, "%s", fruta);
```

```
    printf("ingresó: %s\n", fruta);
```

```
    return 0;
```

```
}
```

```
#include<stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("%s", "banana\n");
```

```
    return 0;
```

```
}
```

```
> gcc -Wall -std=c90 -pedantic-errors archivos-00s.c -o imprime-fruta
> gcc -Wall -std=c90 -pedantic-errors archivos-00t.c -o imprime-banana
> ./imprime-banana
banana
> ./imprime-banana | ./imprime-fruta
ingresó: banana
```



# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```



# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

# Archivos

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    FILE *pf;
```

```
    char fruta[80];
```

```
    pf = fopen("fruta.txt", "r");
```

```
    fscanf(pf, "%s", fruta);
```

```
    printf("%s\n", fruta);
```

```
    fclose(pf);
```

```
    return 0;
```

```
}
```

```
> cat fruta.txt
```

```
naranja
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");
    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("fruta.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat fruta.txt
naranja
```

```
naranja
```





# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("verdura.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("verdura.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat verduras.txt
No existe el fichero o el directorio
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("verdura.txt", "r");

    fscanf(pf, "%s", fruta);

    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat verduras.txt
No existe el fichero o el directorio
```

```
segmentation fault
```



# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("verdura.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    fscanf(pf, "%s", fruta);
    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

# Archivos

```
#include<stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("verdura.txt","r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    fscanf(pf, "%s", fruta);
    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat verdura.txt
No existe el fichero o el directorio
```

# Archivos

```
#include<stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("verdura.txt","r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    fscanf(pf, "%s", fruta);
    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat verdura.txt
No existe el fichero o el directorio
```

```
El archivo no existe.
```





# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    fscanf(pf, "%s", fruta);
    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    fscanf(pf, "%s", fruta);
    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    fscanf(pf, "%s", fruta);
    printf("%s\n", fruta);

    fclose(pf);

    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

```
naranja
```



# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

- `feof` está definida en `stdio.h`



# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

- `feof` está definida en `stdio.h`
- devuelve 0 mientras no se llegue al final del archivo.

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

- `feof` está definida en `stdio.h`
- devuelve 0 mientras no se llegue al final del archivo.
- devuelve 1 cuando se llega al final

# Archivos

```
#include <stdio.h>

int main(void)
{
    FILE *pf;
    char fruta[80];

    pf = fopen("frutas.txt", "r");
    if ( pf == NULL )
    {
        printf("El archivo no existe.\n");
        return 1;
    }

    while ( !feof(pf) ){
        fscanf(pf, "%s ", fruta);
        printf("%s\n", fruta);
    }

    fclose(pf);
    return 0;
}
```

```
naranja
manzana
banana
pera
uva
```

```
> cat frutas.txt
naranja
manzana
banana
pera
uva
```

- `feof` está definida en `stdio.h`
- devuelve 0 mientras no se llegue al final del archivo.
- devuelve 1 cuando se llega al final

claudiojpaz@gmail.com

**Horario de Consulta: Martes 18:00-19:00hs**  
**Of.5 Ed.Salcedo**