

tipus

Mode = (lectura, escriptura)

ftipus

El tipus Cadena és un tipus cadena de caràcters.

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

```
typedef enum {  
    lecturaFST,  
    escripturaFST  
} ModeFST;
```

```
typedef FILE *FST;
```

{Prec: $m = \text{lectura} \Rightarrow \text{existeix}(\text{nom})$ }

funció *ObrirFST*(**ent** $m : \text{Mode}$, **ent** $\text{nom} : \text{Cadena}$) **retorna** *FST*

{Post: $\text{ObrirFST}(m, \text{nom}) = f \wedge$

$(m = \text{lectura} \Rightarrow$

$\text{pe}(f) = \text{en_curs}(f) = \emptyset \wedge \text{pd}(f) = \text{seqüència}(\text{nom})) \wedge$

$(m = \text{escriptura} \Rightarrow \text{pe}(f) = \text{en_curs}(f) = \text{pd}(f) = \emptyset)$ }

```
FST ObrirFST(ModeFST m, const char *nom)
{
    FST f;

    if (m == lecturaFST) {
        f = (FST)fopen(nom, "r");
    } else if (m == escripturaFST) {
        f = (FST)fopen(nom, "w");
    } else assert(0);
    assert(f != NULL);
    return f;
}
```

{Post: $mode(f) = lectura \wedge$

$pe(f) = \alpha \wedge en_curs(f) = \beta \wedge pd(f) = \gamma \neq \emptyset$ }

acció *LlegirCaracterFST*(**entsor** $f : FST$, **sor** $c : \text{caràcter}$)

{Post: $mode(f) = lectura \wedge$

$pe(f) = \alpha.\beta \wedge c = en_curs(f) = cap(\gamma) \wedge pd(f) = cua(\gamma)$ }

```
void LlegirCaracterFST(FST f, char *const c)
{
    *c = fgetc((FILE *)f);
    assert(!feof((FILE *)f) || !ferror((FILE *)f));
}
```

acció *LlegirEnterFST*(**entsor** *f* : *FST*, **sor** *e* : **enter**)

```
void LlegirEnterFST(FST f, int *const e)
{
    int ret;

    ret = fscanf((FILE *)f, "%d", e);
    assert(!feof((FILE *)f) || (ret == 1 && !ferror((FILE *)f)));
}
```

{Prec: $mode(f) = \text{escriptura} \wedge pe(f) = \alpha$ }

acció *EscriureCaracterFST*(**entsor** $f : FST$, **ent** $c : \text{caràcter}$)

{Post: $mode(f) = \text{escriptura} \wedge pe(f) = \alpha.c$ }

```
void EscriureCaracterFST(FST f, char c)
{
    char    ret;
    ret = fputc(c, (FILE *)f );
    assert(ret == c && !ferror((FILE *)f));
}
```

acció *EscriureEnterFST*(**entsor** *f* : *FST*, **ent** *e* : **enter**)

```
void EscriureEnterFST(FST f, int e)
{
    int ret;
    ret = fprintf((FILE *)f, "%d", e);
    assert(ret >= 0 && !ferror((FILE *)f));
}
```

{Post: $mode(f) = lectura$ }

funció $FdFST$ (ent $f : FST$) **retorna booleà**

{Post: $FdFST(f) \equiv sentinella(en_curs(f))$ }

```
int FdFST(FST f)
{
    return feof((FILE *)f);
}
```


{Post: $obert(f)$ }

acció *TancarFST*(entsor $f : FST$)

{Post: $\neg obert(f)$ }

```
void TancarFST(FST f)
{
    int ret;

    ret = fclose((FILE *)f);
    assert(ret == 0);
}
```