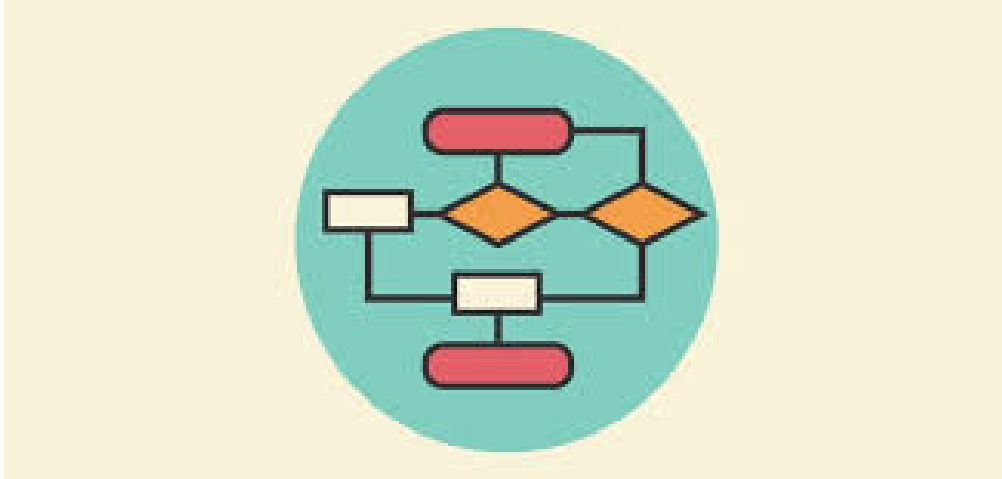


# ALGORITMO



## ÍNDICE

<b>EJERCICIO 1.....</b>	<b>3</b>
a. Tipus dades tPosition.....	3
b. Tipus dades tWord.....	3
c. Tipus dades tSoup.....	3
d. Tipus dades tSearch.....	3
e. Tipus dades tGame.....	4
<b>EJERCICIO 2.....</b>	<b>5</b>
<b>EJERCICIO 3.....</b>	<b>8</b>
- Obtenir lletra aleatòria.....	8
- Acció per emplenar amb lletres aleatòries.....	8
- Sortida de la sopa de lletra.....	8
- Paraula a la sopa de lletres.....	9
<b>EJERCICIO 4.....</b>	<b>10</b>
- Acció reverseWord().....	10
- Modificació exercici 3.....	10

# EXERCICI 1

## a. Tipus dades tPosition

```
tipus
    tLugar { HORIZONTAL, VERTICAL, DIAGONAL}
    tPosition: tupla
        horizontal: entero
        vertical: entero
        diagonal: tLugar
    ftupla
ftipus
```

## b. Tipus dades tWord

```
const
    CHARACTERS: char= 15;
fconst
tipus
    tWord: tupla
        paraula: taula [CHARACTERS] de tPosition;
        posicion: tPosition
        encontrar: boolean
    ftupla
ftipus
```

## c. Tipus dades tSoup

```
tipus
    tEstat {BUIDA, OMLERTA, PARAULA};
    tSoup: tupla
        sopa: tEstat
        paraula: tWord
    ftupla
ftipus
```

## d. Tipus dades tSearch

```
const
    PARAULES: char[] = 15;
fconst
tipus
    tSearch: taula[PARAULES] de tWord;
ftipus
```

### **e. Tipus dades tGame**

tipus

tGame: tupla

sopa: tSoup

paraula\_cercar: tWord

ftupla

ftipus

## EXERCICI 2

funcio getCharacter (sopa : tSoup, fila : sencer, columna : sencer) : caràcter

var

car : caràcter

fvar

si fila >= 1 i fila <= num\_filas(sopa) i columna >= 1 i columna <= num\_columnnes(sopa)

llavors

car := sopa[fila, columna]

retorna car

sinó

retorna

fsi

ffuncio

acció setCharacter (sal sopa : tSoup, fila : sencer, columna : sencer, caràcter : caràcter)

fvar

si fila >= 1 i fila <= num\_filas(sopa) i columna >= 1 i columna <= num\_columnnes(sopa)

llavors

sopa[fila, columna] := caràcter

finsi

facció

acció setWord (sal sopa : tSoup, paraula : tWord, fila : sencer, columna : sencer, orientació : tLloc)

var

i : sencer

longitudParaula : sencer

novaFila, novaColumna : sencer

fvar

longitudParaula := longitud(paraula.paraula)

per a := 1 fins a longitudParaula fer

novaFila := fila

novaColumna := columna

segons orientacio fer

cas horitzontal: novaColumna := columna + i - 1

cas vertical: novaFila := fila + i - 1

cas diagonal:

novaFila := fila + i - 1

novaColumna := columna + i - 1

fi\_segon

si novaFila >= 1 i novaFila <= num\_filas(sopa) i novaColumna >= 1 i novaColumna <= num\_columnnes(sopa) llavors

caracter\_de(paraula.paraula[i])

```
    paraula.paraula[i].horizontal := novaColumna
    paraula.paraula[i].vertical := novaFila
    paraula.paraula[i].diagonal := orientacion
```

finsi

fpara

```
    paraula.trobar := fals
    paraula.posicion.horizontal := columna
    paraula.posicion.vertical := fila
    paraula.posicion.diagonal := orientacion
```

facció

acció initSoup (sal sopa : tSoup, numFiles : sencer, numColumnnes : sencer)

var

i, j : sencer

fvar

redimensionar(sopa.sopa, numFiles, numColumnas) // Funció hipotètica

```
    per a i := 1 fins numFiles fer
    per j := 1 fins numColumnnes fer
    sopa.sopa[i, j] := vacia
    fpara
    fpara
    facció
```

funcio readWord () : tWord

var

```
    paraulaLeida : array [1..CHARACTERS] DE caracter
    longitudLeida : sencer
    i : sencer
    novaParaula : tWord
```

fvar

llegir\_cadena(paraulaLeida) // Funció hipotètica

longitudLeida := longitud(paraulaLeida)

```
    per a i := 1 fins a CHARACTERS fer
    novaParaula.paraula[i].horitzontal := 0
    novaParaula.paraula[i].vertical := 0
    novaParaula.paraula[i].diagonal := horitzontal
    fpara
    novaParaula.posicio.horitzontal := 0
    novaParaula.posicion.vertical := 0
    novaParaula.posicion.diagonal := horitzontal
    novaParaula.trobar := FALS
```

torna novaParaula  
ffuncio

## EXERCICI 3

### - Obtenir lletra aleatòria

```
funcio getRandomCharacter(): caracter
var
    min, max, code: enter;
fvar
    min := 65;
    max := 90;
    code := getRandomNumber(min, max);
    retorna chr(code);
ffuncio
```

### - Acció per emplenar amb lletres aleatòries

```
accio fillWithRandomCharacters(ent n: enter, ent m: enter, entsor soup: tSoup)
var
    c: caracter;
    i, j: enter;
fvar
    per i := 1 fins n fer
        per j := 1 fins m fer
            c := getRandomCharacter();
            setSoupChar(i, j, c, soup);
        fper
    fper
faccio
```

### - Sortida de la sopa de lletra

```
accio writeSoup(ent n,m: enter, ent soup: tSoup)
var
    i, j: enter;
    c: caracter;
fvar
    write(n); write(' '); write(m);
    per i := 1 fins n fer
        per j := 1 fins m fer
            c := getSoupChar(i, j, soup);
```



```

        write(' ');
        writeChar(c);
    fper
fper
faccio

```

## - Paraula a la sopa de lletres

```

accio hideWordIntoSoup(entsor word: tWord, ent n: enter, ent m: enter, entsor soup: tSoup)
var
    i, j, k, ori, len: enter;
    c: caracter;
fvar
    len := getWordLength(word);
    ori := getRandomNumber(1, 3);

    si ori = 1 llavors
        i := getRandomNumber(1, n - len + 1);
        j := getRandomNumber(1, m);
        per k := 1 fins len fer
            c := getCharacterAt(word, k);
            setSoupChar(i + k - 1, j, c, soup);
        fper
    sino si ori = 2 llavors
        i := getRandomNumber(1, n);
        j := getRandomNumber(1, m - len + 1);
        per k := 1 fins len fer
            c := getCharacterAt(word, k);
            setSoupChar(i, j + k - 1, c, soup);
        fper
    sino
        i := getRandomNumber(1, n - len + 1);
        j := getRandomNumber(1, m - len + 1);
        per k := 1 fins len fer
            c := getCharacterAt(word, k);
            setSoupChar(i + k - 1, j + k - 1, c, soup);
        fper
    fsi
faccio

```

## EXERCICI 4

### - Acció reverseWord()

```
accio reverseWord(entsor word: tWord)
var
  i, j: enter;
  tmp: caracter;
fvar
  i := 1;
  j := getWordLength(word);
  mentre i < j fer
    tmp := getCharacterAt(word, i);
    setCharacterAt(word, i, getCharacterAt(word, j));
    setCharacterAt(word, j, tmp);
    i := i + 1;
    j := j - 1;
  fmentre
faccio
```

### - Modificació exercici 3

```
per i:= 1 fins w fer
  word := readWord();

  si getRandomNumber(0,1) = 1 llavors
    reverseWord(word);
  fsi

  addWordToSearch(search, word);
  hideWordIntoSoup(word, n, m, soup);
fper
```