



UNIVERSIDAD SIMÓN  
BOLÍVAR

INFORME DE PROYECTO I

---

# Estructuras de Datos en MIPS

---

**Autor:**

Christopher Gómez  
Ka Fung

**Profesor:**

Eduardo Blanco

**Organización del Computador (CI3815)**

24 de noviembre de 2021

## 1. Pseudocódigo

Se presentan a continuación los pseudocódigos del programa que se desea implementar.

Primeramente, se necesita extraer los datos necesarios de cada archivo de entrada, para ello, se usan tablas de hash y listas.

### EXTRAER-DATOS

```
1  archivoEst = Leer archivo de estudiantes
2  tablaHashEst = new TABLAHASH(101)
3  for linea in archivoEst:
4      carnet = Guardar carnet
5      nombre = Guardar nombre
6      indice = Guardar índice
7      creditosAprob = Guardar número de créditos aprobados
8      est = new ESTUDIANTE(carnet, nombre, indice, creditosAprob)
9      tablaHashEst.INSERTAR(carnet, est)
10
11 archivoMat = Leer archivo de materias
12 tablaHashMat = new TABLAHASH(101)
13 listaMat = new LISTA()
14 for linea in archivoMat:
15     codigo = Guardar codigo
16     nombre = Guardar nombre
17     creditos = Guardar creditos
18     numCupos = Guardar número de cupos
19     minCreditos = Guardar mínimo de créditos
20     mat = new MATERIA(codigo, nombre, creditos, numCupos, minCreditos)
21     tablaHashMat.INSERTAR(carnet, est)
22     listaMat.INSERTAR-ORDENADO(codigo)
23
24 listaSol = new LISTA()
25 archivoSol = Leer archivo de solicitudes
26 for linea in archivoSol:
27     carnet = Guardar carnet del estudiante
28     est = tablaHashEst.OBTENER-VALOR(carnet)
29     codigo = Guardar codigo de la materia
30     mat = tablaHashMat.OBTENER-VALOR(codigo)
31     sol = new SOLICITUD(est, mat)
32     listaSol.INSERTAR(sol)
```

Al terminar este pseudocódigo, se debe tener una lista de solicitudes, una lista de códigos de materias en orden lexicográfico, una tabla de estudiantes, y una tabla de materias, la idea ahora es procesar la lista de solicitudes para que cada materia tenga una lista de estudiantes inscritos.

### PROCESAR-SOLICITUDES

```
33 for sol in listaSol:
34     est = sol.estudiante
35     mat = sol.materia
36     mat.AGREGAR-ESTUDIANTE(est)
```

Ahora, cada materia contiene una lista con los estudiantes inscritos. Se supone que la estructura se encarga de mantener actualizado el número de cupos y de agregar en orden a los estudiantes en su lista de estudiantes. Así, para finalizar esta primera etapa solamente resta escribir en el archivo de salida cada materia con sus estudiantes inscritos.

#### GENERAR-ARCHIVO-TENTATIVO

```
37 archivoTen = Abrir archivo tentativo a escribir
38 for mat in listaMat:
39     archivoTen.ESCRIBIR('<mat.codigo> ')
40     archivoTen.ESCRIBIR("<mat.nombre>" ')
41     archivoTen.ESCRIBIR('<mat.numCupos> \n')
42     for est in mat.estudiantes:
43         archivoTen.ESCRIBIR('<est.carnet> ')
44         archivoTen.ESCRIBIR('<est.nombre> \n')
```

## 2. Estructuras utilizadas

En la seccion anterior se menciona el uso de distintas estructuras de datos utilizadas en el diseño del programa. En esta sección se describe cada una de ellas, junto con sus atributos y operaciones.

- LISTA:
  - Atributos:
    - Cabeza.
    - Tamaño.
  - Operaciones:
    - CREAM()