



OBJETIVO

Este documento contiene material de apoyo para el boletín T14 del trabajo práctico. En concreto, el documento incluye el código en lenguaje C para implementar y testear los tipos Cadena, Asignatura y Grado. El profesor le explicará estas soluciones en la sesión de laboratorio y responderá sus dudas al respecto. El alumno, por su parte, debe seguir los pasos del enunciado para crear el proyecto C, añadir el código aquí mostrado y realizar por su cuenta el resto de ejercicios del boletín.

SESIÓN 1

Archivo `cadena.h`

```
#ifndef CADENA_H_
#define CADENA_H_

#define MAXCAR 256
typedef char Cadena[MAXCAR];
void quitaSaltoDeLinea(Cadena c);

#endif /* CADENA_H_ */
```

Archivo `cadena.c`

```
#include <string.h>
#include "cadena.h"

void quitaSaltoDeLinea(Cadena c) {
    int len=strlen(c);
    if(c[len-1]=='\n') {
        c[len-1]='\0';
    }
}
```

Archivo `logico.h`

```
#ifndef LOGICO_H_
#define LOGICO_H_

typedef enum {FALSO, CIERTO} Logico;

#endif /* LOGICO_H_ */
```



Archivo asignatura.h

```
#ifndef ASIGNATURA_H_
#define ASIGNATURA_H_

#include "cadena.h"
#include "logico.h"

#define MAXASIG 100
#define CODIGOLEN 8

typedef char Codigo[CODIGOLEN];

typedef struct{
    Cadena nombre;
    Codigo codigo;
    double credits;
    int curso;
    Cadena departamento;
} Asignatura;

typedef Asignatura *PAsignatura;
typedef Asignatura ArrayAsignaturas[MAXASIG];

int inicializaAsignatura(PAsignatura res, const Cadena nombre, const Codigo
codigo, double credits, int curso, const Cadena departamento);
void muestraAsignatura(Asignatura a);
void muestraAsignaturas(const ArrayAsignaturas res, int nAsig);

/* Funciones auxiliares */
Logico checkCodigo(const Codigo codigo);
Logico checkCurso(int curso);
Logico checkCredits(double credits);

#endif /* ASIGNATURA_H_ */
```

Archivo asignatura.c

```
#include "asignatura.h"
#include <string.h>

int inicializaAsignatura(PAsignatura res, const Cadena nombre, const Codigo
codigo, double credits, int curso, const Cadena departamento) {
    int status = 0;

    if (!checkCodigo(codigo)) {
        fprintf(stderr, "Error en código de asignatura: %s\n", codigo);
        status = -1;
    }
    else if (!checkCredits(credits)) {
        fprintf(stderr, "Error en créditos de asignatura: %lf\n",
credits);
        status = -1;
    }
}
```

Archivo asignatura.c (continúa)

```
        else if (!checkCurso(curso)) {
            fprintf(stderr, "Error en curso de asignatura: %d\n", curso);
            status = -1;
        }
        else {
            strcpy(res->nombre, nombre);
            strcpy(res->codigo, codigo);
            res->curso = curso;
            res->creditos = creditos;
            strcpy(res->departamento, departamento);
        }
        return status;
    }

void muestraAsignatura(Asignatura a) {
    printf("Nombre: %s\n", a.nombre);
    printf("Código: %s\n", a.codigo);
    printf("Créditos: %lf\n", a.creditos);
    printf("Curso: %d\n", a.curso);
    printf("Departamento: %s\n", a.departamento);
}

void muestraAsignaturas(const ArrayAsignaturas res, int nAsig) {
    int i;

    for (i = 0; i < nAsig; i++) {
        printf("Asignatura número %d.\n", i + 1);
        muestraAsignatura(res[i]);
        printf("=====\n");
    }
}

Logico checkCodigo(const Codigo codigo) {
    Logico res = FALSO;
    if(strlen(codigo) == CODIGOLEN-1){
        res = CIERTO;
    }
    return res;
}

Logico checkCurso(int curso) {
    Logico res = FALSO;
    if(curso >= 1 && curso <= 4){
        res = CIERTO;
    }
    return res;
}

Logico checkCreditos(double creditos) {
    Logico res = FALSO;
    if(creditos>0.0){
        res = CIERTO;
    }
    return res;
}
```

SESIÓN 2

Archivo `asignatura.h` (añadir las siguientes declaraciones)

```
#ifndef ASIGNATURA_H_
#define ASIGNATURA_H_

#include <stdio.h> /* Para usar FILE */
(...)
int leeAsignaturasFichero(const Cadena nombreFichero, ArrayAsignaturas res);
(...)
/* Funciones auxiliares */
(...)
void leeAsignaturaFichero(FILE *f, PAsignatura a);
```

Archivo `asignatura.c` (añadir las siguientes funciones)

```
int leeAsignaturasFichero(const Cadena nombreFichero, ArrayAsignaturas res) {
    int i = 0;
    int n;
    FILE *f;

    f = fopen(nombreFichero, "r");
    if (f == NULL) {
        printf("Fichero %s no encontrado", nombreFichero);
        n=0;
    } else {
        leeAsignaturaFichero(f,&res[i]);
        while (!feof(f) && i < MAXASIG - 1) {
            i++;
            leeAsignaturaFichero(f,&res[i]);
        }
        n = i;
        fclose(f);
    }
    return n;
}

void leeAsignaturaFichero(FILE *f, PAsignatura a){
    char c;

    fgets(a->nombre, MAXCAR, f);
    quitaSaltoDeLinea(a->nombre);
    fgets(a->codigo, CODIGOLEN, f);
    quitaSaltoDeLinea(a->codigo);
    fscanf(f, "%lf%c", &a->creditos, &c);
    fscanf(f, "%d%c", &a->curso, &c);
    fgets(a->departamento, MAXCAR, f);
    quitaSaltoDeLinea(a->departamento);
}
```



Archivo grado.h

```
#ifndef GRADO_H_
#define GRADO_H_

#include "asignatura.h"
#include "cadena.h"
#include "logico.h"

typedef struct {
    Cadena nombre;
    Cadena centro;
    ArrayAsignaturas obligatorias;
    int numObligatorias;
    ArrayAsignaturas optativas;
    int numOptativas;
    double minimoCreditosOptativas;
} Grado;

typedef Grado *PGrado;

int inicializaGrado(PGrado res, const Cadena nombre, const Cadena centro,
const ArrayAsignaturas obligatorias, int numObligatorias, const
ArrayAsignaturas optativas, int numOptativas, double minimoCreditosOptativas);
void muestraGrado(Grado g);

/* Funciones auxiliares */
Logico checkMinimoCreditosOptativas(double minimoCreditosOptativas,
const ArrayAsignaturas optativas, int numOptativas);
Logico checkOptativasTodasIgualesCreditos(const ArrayAsignaturas optativas,
int numOptativas);
double sumaCreditos(const ArrayAsignaturas optativas, int numOptativas);
void copiaArray(ArrayAsignaturas destino, const ArrayAsignaturas origen, int
numElem);

#endif /* GRADO_H_ */
```

Archivo grado.c

```
#include <stdio.h>
#include <string.h>
#include "grado.h"

int inicializaGrado(PGrado res, const Cadena nombre, const Cadena centro,
const ArrayAsignaturas obligatorias, int numObligatorias,
const ArrayAsignaturas optativas, int numOptativas,
double minimoCreditosOptativas) {
    int status = 0;

    if (!checkMinimoCreditosOptativas(minimoCreditosOptativas, optativas,
numOptativas)) {
        fprintf(stderr, "Error con el número de créditos mínimo de las
asignaturas optativas.\n");
        status = -1;
    }
}
```

Archivo grado.c (continúa)

```
        else if (!checkOptativasTodasIgualesCreditos(optativas, numOptativas)) {
            fprintf(stderr, "Error con el número de créditos de las
asignaturas optativas.\n");
            status = -1;
        }
        else{
            strcpy(res->nombre, nombre);
            strcpy(res->centro, centro);
            copiaArray(res->obligatorias, obligatorias, numObligatorias);
            res->numObligatorias = numObligatorias;
            copiaArray(res->optativas, optativas, numOptativas);
            res->numOptativas = numOptativas;
            res->minimoCreditosOptativas = minimoCreditosOptativas;
        }

        return status;
    }

void muestraGrado(Grado g){
    printf("Nombre: %s\n", g.nombre);
    printf("Centro: %s\n", g.centro);
    printf("Mínimo créditos optativas: %lf\n",g.minimoCreditosOptativas);
    printf("Asignaturas obligatorias:\n");
    muestraAsignaturas(g.obligatorias,g.numObligatorias);
    printf("Asignaturas optativas:\n");
    muestraAsignaturas(g.optativas,g.numOptativas);
}

Logico checkMinimoCreditosOptativas(double minimoCreditosOptativas,
    const ArrayAsignaturas optativas, int numOptativas) {
    Logico res = FALSO;
    double suma = sumaCreditos(optativas, numOptativas);
    if(minimoCreditosOptativas >= 0 && minimoCreditosOptativas <= suma){
        res = CIERTO;
    }
    return res;
}

double sumaCreditos(const ArrayAsignaturas optativas, int numOptativas) {
    double suma = 0.0;
    int i;

    for (i = 0; i < numOptativas; i++) {
        suma += optativas[i].creditos;
    }
    return suma;
}
```



Archivo grado.c (continúa)

```
Logico checkOptativasTodasIgualesCreditos(const ArrayAsignaturas optativas,
                                           int numOptativas) {
    Logico res = CIERTO;
    int i;

    for(i=1; i<numOptativas; i++) {
        if(optativas[i].creditos != optativas[0].creditos) {
            res = FALSO;
            break;
        }
    }
    return res;
}

void copiaArray(ArrayAsignaturas destino, const ArrayAsignaturas origen, int
numElem) {
    int i;

    for(i=0; i<numElem; i++) {
        destino[i] = origen[i];
    }
}
```