

PROYECTO FINAL / IIS

**Asignatura: Introducción a la Ingeniería de
Software**

**UACM SAN LORENZO
TEZONCO**

**ROSAS HERNANDEZ KEVIN
YOHEL / 23-003-0286**

Fecha de entrega: 11 de diciembre de 2025

**Repositorio GitHub:
<https://github.com/Yohelpch7/proyecto-final-iis.git>**

Descripción

El sistema de gestión escolar ofrece la posibilidad de crear cursos, registrar alumnos, inscribirlos en las materias correspondientes, consultar las listas de inscritos, calcular el promedio de cada curso, generar boletas individuales y determinar cuál curso tiene el mayor número de estudiantes aprobados. Está desarrollado en lenguaje C y utiliza memoria dinámica tanto para el manejo del arreglo de cursos como para los alumnos asociados a cada curso. Además, guarda su información en un archivo binario (cursos.dat) para asegurar la persistencia de los datos entre ejecuciones.

Objetivo general

Construir un sistema sencillo que administre el registro de cursos y estudiantes, garantizando la persistencia de la información y el uso adecuado de memoria dinámica.

Alcance

Las funciones principales incluyen:

- ❖ Creación de cursos
 - ❖ Registro de estudiantes
 - ❖ Inscripción de alumnos en cursos
 - ❖ Cálculo de promedios por curso
 - ❖ Generación de boletas
 - ❖ Reporte del curso con mayor número de aprobados
- La interfaz es de tipo consola – texto- y la persistencia de datos se logra mediante archivos binarios.

Registros PSP:

Fase	Fecha de Inicio	Fecha de Fin	Tiempo Total (min)	Observaciones
Planeación	08/Dic/2025	08/Dic/2025	35	Definición de estructuras
Diseño	09/Dic/2025	09/Dic/2025	82	Anclacion de las 10 funciones y el flujo del menú.
Codificación	09/Dic/2025	10/Dic/2025	145	Implementación de funciones, persistencia y lógica.
Pruebas	10/Dic/2025	10/Dic/2025	42	Desarrollo de la función pruebas y verificación manual.
Documentación	10/Dic/2025	10/Dic/2025	90	Creación de Reporte PDF
Total General			394 min o 6 hrs con 34 min	

funciones

cargarDatos()

Parámetros de entrada:
Ninguno (lee de ARCHIVO_CURSOS)

Parámetros de Salida:
Ninguno (modifica variables globales)

funcionalidad:
Carga el estado previo del sistema desde el archivo binario. Lee el total de cursos, y luego, para cada curso, asigna memoria dinámica (malloc) para sus alumnos y carga los datos.

guardarDatos()

Parámetros de entrada:
Ninguno (escribe a ARCHIVO_CURSOS)

Parámetros de Salida:
Ninguno

funcionalidad:
Guarda el estado actual del sistema en el archivo binario. Escribe el total de cursos y la información de cada curso junto con el bloque de datos de sus alumnos inscritos.

registrarAlumno()

Parámetros de entrada:
Datos del alumno (Matrícula, Nombres, Apellidos) por scanf

Parámetros de Salida:
Ninguno

funcionalidad:
Agrega un nuevo alumno a la lista maestra estática (registrados), verificando previamente que la matrícula no exista.

altaCurso()

Parámetros de entrada:
Clave, Nombre, Profesor del curso por scanf

Parámetros de Salida:
Ninguno

funcionalidad:
Crea una nueva estructura de curso. Asigna memoria dinámica (malloc) para los 15 cupos de alumnos que tendrá dicho curso, cumpliendo el requisito de memoria dinámica.

inscribirAlumnoCurso()

Parámetros de entrada:
Clave del curso, Matrícula del alumno por scanf

Parámetros de Salida:
Ninguno

funcionalidad:
Inscribe a un alumno en un curso. Valida que el alumno esté previamente registrado y que el curso no exceda el cupo máximo de 15 estudiantes.

listarAlumnosCurso()

Parámetros de entrada:
Clave del curso por scanf.

Parámetros de Salida:
Ninguno

funcionalidad:
Muestra la lista de estudiantes inscritos en el curso especificado, incluyendo su calificación actual.

promedioCurso()

Parámetros de entrada:
Clave del curso por scanf

Parámetros de Salida:
Ninguno

funcionalidad:
Calcula y muestra el promedio de las calificaciones de todos los alumnos inscritos en el curso.

boletaAlumno()

Parámetros de entrada:
Matrícula del alumno por scanf

Parámetros de Salida:
Ninguno

funcionalidad:
Genera la boleta del estudiante. Recorre todos los cursos y muestra el nombre del curso, la calificación obtenida, y el estado (Aprobado/Reprobado) basado en el mínimo aprobatorio de 7.

funciones

cursoConMasAprobados()

Parámetros de entrada:

Ninguno

Parámetros de Salida:
Ninguno

funcionalidad:

Determina, entre todos los cursos registrados, cuál tiene el mayor número de alumnos con calificación < 7

pruebas()()

Parámetros de entrada:

Ninguno

Parámetros de Salida:
Ninguno

funcionalidad:

Función de control diseñada para incluir y ejecutar el código de prueba unitario de las demás funciones, según los lineamientos del proyecto.

Algoritmo de la función principal (diagrama de flujo o seudocódigo):

main

Inicio

reservar memoria para arreglo dinámico cursos (capacidad max_cursos)

llamar a cargarDatos()

repetir

mostrar menú

leer opción del usuario

según opción hacer

1 > registrarAlumno()

2 > altaCurso()

3 > inscribirAlumnoCurso()

4 > listarAlumnosCurso()

5 > promedioCurso()

6 > boletaAlumno()

7 > cursoConMasAprobados()

8 > pruebas()

o > guardarDatos()

mientras opción != o

liberar memoria dinámica de cada cursos[i].alumnos

liberar memoria del arreglo cursos

Fin

Enlace del repositorio:

<https://github.com/Yohelpch7/proyecto-final-iis.git>

Capturas de pantalla:

```
main.c
396     int aprob = 0;
397
398     for (int j = 0; j < cursos[i].total_inscritos; j++) {
399         if (cursos[i].alumnos[j].calificacion >= MIN_APROBATORIO)
400             aprob++;
401
402     }
403
404     if (aprob > maxA) {
405         maxA = aprob;
406         mejor = i;
407     }
408
409
410     if (mejor == -1) {
411         printf("No hay aprobados.\n");
412         return;
413     }
414
415     printf("Curso con más aprobados: %s (clave %d)\n",
416           cursos[mejor].nombre,
417           cursos[mejor].clave);
418 }
419
420 // 10) pruebas
421 void pruebas() {
422     printf("Pruebas simples:\n");
423     printf("- Prueba registrar: crear alumno nuevo\n");
424     printf("- Prueba alta curso: crear curso y luego listar\n");
425     printf("- Prueba inscribir: registrar alumno y meterlo a curso\n");
426     printf("(Solo prints, versión sencilla.)\n");
427 }
```

Opción:
Número: 23-223-2569
Nombre: Raul
Apellido paterno: Reyes
Apellido materno: Sanchez
Alumno registrado.

== SISTEMA ESCOLAR ==
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso

```
main.c
396     int aprob = 0;
397
398     for (int j = 0; j < cursos[i].total_inscritos; j++) {
399         if (cursos[i].alumnos[j].calificacion >= MIN_APROBATORIO)
400             aprob++;
401
402     }
403
404     if (aprob > maxA) {
405         maxA = aprob;
406         mejor = i;
407     }
408
409
410     if (mejor == -1) {
411         printf("No hay aprobados.\n");
412         return;
413     }
414
415     printf("Curso con más aprobados: %s (clave %d)\n",
416           cursos[mejor].nombre,
417           cursos[mejor].clave);
418 }
419
420 // 10) pruebas
421 void pruebas() {
422     printf("Pruebas simples:\n");
423     printf("- Prueba registrar: crear alumno nuevo\n");
424     printf("- Prueba alta curso: crear curso y luego listar\n");
425     printf("- Prueba inscribir: registrar alumno y meterlo a curso\n");
426     printf("(Solo prints, versión sencilla.)\n");
427 }
```

Opción:
Número: 23-223-2569
Nombre: Raul
Apellido paterno: Reyes
Apellido materno: Sanchez
Alumno registrado.

== SISTEMA ESCOLAR ==
. Registrar estudiante
. Dar de alta curso
. Inscribir estudiante a curso
. Lista de estudiantes por curso
. Promedio de un curso
. Boleta de estudiante
. Curso con más aprobados
. Ejecutar pruebas
. Guardar y salir

Opción: ■

```
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 6
Matrícula: 23-003-0286
Estudiante no registrado.

==== SISTEMA ESCOLAR ====
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 0
Datos guardados.

...Program finished with exit code 0
Press ENTER to exit console.
```

```
main.c
395:     for (int i = 0; i < cursos.size(); i++) {
396:         int aprob = 0;
397:
398:         for (int j = 0; j < cursos[i].total_inscritos; j++) {
399:             if (cursos[i].alumnos[j].calificacion >= MIN_APROBATORIO)
400:                 aprob++;
401         }
402:
403:         if (aprob > maxA) {
404:             maxA = aprob;
405:             mejor = i;
406         }
407     }
408
409     if (mejor == -1) {
410         printf("No hay aprobados.\n");
411     }
412     return;
413 }
```

```
0. Guardar y salir
Opción: 7
No hay cursos.

==== SISTEMA ESCOLAR ====
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 6
Matrícula: 23-003-0286
Estudiante no registrado.

==== SISTEMA ESCOLAR ====
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 
```

```
main.c
396     for (int i = 0; i < cursos.size(); i++) {
397         int aprobo = 0;
398
399         for (int j = 0; j < cursos[i].total_inscritos; j++) {
400             if (cursos[i].alumnos[j].calificacion >= MIN_APROBATORIO)
401                 aprobo++;
402
403             if (aprob > maxA) {
404                 maxA = aprobo;
405                 mejor = i;
406             }
407         }
408
409         if (mejor == -1) {
410             printf("No hay aprobados.\n");
411             return;
412         }
413     }
0. Guardar y salir
Opción: 5
Clave del curso: 26793
Curso no encontrado.

==== SISTEMA ESCOLAR ====
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 7
No hay cursos.
```



```
main.c
396     for (int i = 0; i < cursos.size(); i++) {
397         int aprobo = 0;
398
399         for (int j = 0; j < cursos[i].total_inscritos; j++) {
400             if (cursos[i].alumnos[j].calificacion >= MIN_APROBATORIO)
401                 aprobo++;
402
403             if (aprob > maxA) {
404                 maxA = aprobo;
405                 mejor = i;
406             }
407         }
408
409         if (mejor == -1) {
410             printf("No hay aprobados.\n");
411             return;
412         }
413     }
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 1
Matrícula: 23-223-2569
Nombre: Raul
Apellido paterno: Reyes
Apellido materno: Sanchez
Alumno registrado.

==== SISTEMA ESCOLAR ====
1. Registrar estudiante
2. Dar de alta curso
3. Inscribir estudiante a curso
4. Lista de estudiantes por curso
5. Promedio de un curso
6. Boleta de estudiante
7. Curso con más aprobados
8. Ejecutar pruebas
0. Guardar y salir
Opción: 5
Clave del curso: 26793
Curso no encontrado.

==== SISTEMA ESCOLAR ====

```