

1. **Objetivo General**

- Desarrollar una aplicación que permita reafirmar el conocimiento del lenguaje Ensamblador 8086.

2. **Objetivos Específicos**

- Diseñar e implementar un **sistema interactivo** en lenguaje ensamblador 8086.
- Utilizar las instrucciones de impresión en pantalla.
- Utilizar las instrucciones de lectura del teclado.
- Utilizar las operaciones matemáticas básicas.

3. **Datos Generales**

- El valor de la Tarea: 10%
- **Nombre código:** RegistroCE
- La tarea debe ser implementada en grupos de no más de 3 personas.
- La **fecha de entrega** es 4/Setiembre/2025.
- Cualquier indicio de copia será calificado con una nota de 0 y será procesado de acuerdo con el reglamento.
- La entrega se realizará por la plataforma TEC Digital en su sección correspondiente.

4. **Descripción del caso.**

Se debe desarrollar un **sistema interactivo de gestión de calificaciones en ensamblador 8086**, con las siguientes características mínimas:

4.1. **Menú principal interactivo (texto en pantalla):**

4.1.1. Ingresar calificaciones (hasta 15 estudiantes -Nombre Apellido1 Apellido2 Nota-).

La nota puede ser un número flotante 92.77459

La nota puede ser un número flotante 80.80

La cantidad de decimales máxima es de 5.

4.1.2. Mostrar estadísticas.

4.1.3. Buscar estudiante por posición (índice).

4.1.4. Ordenar calificaciones (ascendente/descendente).

4.1.5. Salir.

#### 4.2. Estadísticas a calcular:

- 4.2.1. Promedio general.
- 4.2.2. Nota máxima y mínima.
- 4.2.3. Cantidad y porcentaje de estudiantes aprobados ( $\geq 70$ ).
- 4.2.4. Cantidad y porcentaje de reprobados ( $< 70$ ).

#### 4.3. Funciones técnicas obligatorias:

- 4.3.1. Implementar subrutinas modulares (CALL y RET).
- 4.3.2. Usar ciclos y comparaciones (LOOP, CMP, Jxx).
- 4.3.3. Implementar un algoritmo de ordenamiento (Burbuja o Selección) sobre el arreglo de calificaciones.
- 4.3.4. Manejo de entrada y salida de: caracteres, números usando interrupciones de DOS (INT 21h).

#### 4.4. Requeimientos NO funcionales

- 4.4.1. Implementar manejo básico de errores (ej: evitar ingresar notas fuera del rango 0–100).

### 5. Entregables

- 5.1. Código fuente comentado.
- 5.2. Manual de usuario.
- 5.3. Documentación Técnica.

### 6. Documentación técnica

1. Se deberá entregar un documento que contenga:
  - 1.1. Descripción detallada del (los) algoritmo(s) de solución desarrollado(s) (con su respectivo(s) diagrama(s)) y su justificación de uso (considerar aspectos de rendimiento).
  - 1.2. Descripción de los procedimientos/funciones implementadas.
  - 1.3. Descripción de la ejemplificación de las estructuras de datos desarrolladas.
  - 1.4. Problemas Encontrados: En esta sección se detalla cualquier problema que se ha presentado en el trabajo y sus soluciones encontradas con su descripción detallada, recomendaciones, conclusiones y bibliografía consultada para este problema específico.
  - 1.5. Problemas conocidos sin solución encontrados: descripción detallada, intentos de solución sin éxito y pasos extra a la aplicación a seguir para solventar el problema.
  - 1.6. Plan de Actividades realizadas por estudiante: Este es un **planeamiento de las actividades que se realizarán para completar la tarea**, este debe incluir descripción de la tarea, tiempo estimado de completitud, responsable a cargo y fecha de entrega.
  - 1.7. Conclusiones.
  - 1.8. Recomendaciones.
  - 1.9. Bibliografía consultada en todo el proyecto.
2. Bitácora en digital, donde se describen las actividades realizadas, desde reuniones con el compañero de trabajo, investigaciones, consultas, etc. Esta se puede encontrar hecha a mano,

se debe describir todo por más insignificante que sea, esto demostrará si ustedes están trabajando en realidad. Este es su diario de trabajo, llevan seguimiento de todo en el tiempo, imaginen que, si un compañero los releva en su trabajo, le bastaría con leer sus bitácoras para seguir el trabajo.

## 7. Evaluación

1. El código tendrá un valor de un 70% de la nota final, debe estar funcional.
2. La documentación tendrá un valor de un 20% de la nota final, cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
3. La defensa tendrá un valor de 10%, todos los integrantes del grupo deben participar. Prepararse para la defensa **puede ser** una pequeña presentación 2-3 diapositivas con la explicación del algoritmo de solución, y los ítems de la rúbrica de evaluación.
4. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código, Documentación y Defensa.
5. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que, aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma imperativo, calidad de documentación interna y externa, plan de actividades y control de plagio con herramientas de IA.
6. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
7. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
8. De las notas mencionadas en el punto 4 se calculará la Nota Final del Proyecto.
9. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
10. Aun cuando el código, la documentación y la defensa tienen sus notas por separado, se aplican las siguientes restricciones
  - 10.1. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
  - 10.2. Si no se entrega el punto 1 de la documentación se obtiene una nota de 0.
  - 10.3. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
  - 10.4. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
  - 10.5. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de 0.
  - 10.6. **NO** presentarse a la defensa se obtendrá una nota de 0.
11. Cada grupo tendrá como máximo 30 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
12. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
13. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
14. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros

profesores y el director de la escuela.

15. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 14, no pueden participar en la revisión.
16. Después de enviada la nota final del proyecto el estudiante tendrá un máximo de 3 días hábiles para presentar un reclamo siempre y cuando la funcionalidad esté completa.