



# **Algoritmos y Estructuras de Datos (1ACC0182)**

## **Trabajo 1**

### **2025-1**

Profesores:	ARIAS ORIHUELA, JOHN EDWARD ARROYO TABOADA, ANGEL DAVID DIAZ SUAREZ, JORGE EDUARDO MENDIOLAZA CORNEJO, EDSON DUILIO ROSALES HUAMANCHUMO, JAVIER ULISES SALAS ARBAIZA, CESAR ENRIQUE SONCCO ALVAREZ, JOSE LUIS SOPLA MASLUCÁN, ABRAHAM
Secciones:	38, 39, 40, 41, 42, 43, 56, 60, 68, 75, 83, 94, 128, 132
Fecha de entrega:	11 de mayo 2025
Ciclo Académico	<b>2025-1</b>

### **1. Objetivo:**

El presente documento define el trabajo parcial y la rúbrica que permite evaluar el logro del curso 1ACC0182 ALGORITMOS Y ESTRUCTURAS DE DATOS. El Trabajo 1 (TB1), tiene por objetivo que los estudiantes apliquen los conocimientos adquiridos en análisis y diseño de tipos de datos abstractos, técnicas sistemáticas con base en matemáticas para la construcción de una aplicación basada en el procesamiento de grandes cantidades de datos, considerando el tiempo de respuesta creativamente.

### **2. Logro del curso:**

Al finalizar el curso, el estudiante construye aplicaciones para el manejo de grandes cantidades de datos de manera ingeniosa.

En Ingeniería de Software, el logro contribuye a alcanzar el:

**ABET – EAC - Student Outcome 2:** La capacidad de aplicar el diseño de ingeniería para producir soluciones que satisfagan necesidades específicas con consideración de salud pública, seguridad y bienestar, así como factores globales, culturales, sociales, ambientales y económicos.

En Ciencias de la Computación, el logro contribuye a alcanzar el:

**ABET – CAC - Student Outcome 2:** Diseñar, implementar y evaluar una solución basada en la computación para cumplir con un conjunto de requisitos computacionales en el contexto de la disciplina del programa.

### **3. Enunciado del Trabajo 1 (TB1)**

El Trabajo 1 del curso de Algoritmos y Estructuras de Datos consiste en implementar una aplicación minimalista con todos los conceptos utilizados en las sesiones de clase de acuerdo al sílabo. La Tabla 1 muestra los grupos, la aplicación a desarrollar por cada grupo y la referencia de ejemplo que pueden desarrollar, este ejemplo sirve para que puedan guiarse en los procesos que tiene cada proyecto.

**Tabla 1: Aplicaciones**

<b>Grupo</b>	<b>Aplicación</b>	<b>Referencia de ejemplo</b>
1	Plataforma de pedidos y delivery	Rappi
2	Sistema de gestión de supermercados	Tottus
3	Aplicación de educación virtual:	Coursera
4	Sistema de transporte urbano	Uber
5	Aplicación de música en streaming	Spotify
6	Sistema de gestión de gimnasios	Smart Fit
7	Sistema de alquiler de vehículos	Localiza
8	Sistema de compras tickets aéreos	JetSmart
9	Sistema de gestión de bibliotecas	Open Library
10	Sistema de gestión de eventos y espectáculos	Teleticket

Para el presente Trabajo 1, se considera el diseño donde la interacción con el usuario debe ser amigable y entendible, se debe cumplir un conjunto de requisitos.

#### **4. Consideraciones generales:**

- A. El proyecto debe ser desarrollado bajo el paradigma de Programación Orientada a Objetos (POO) y de Programación Genérica, requiere al menos 5 entidades por cada integrante de grupo y deben ser consideradas con sus respectivas relaciones entre clases, adjuntar su diagrama de clases UML.
- B. Todas las estructuras de datos desarrolladas en la aplicación deben ser genéricas (implementadas con **templates**) y hacer uso de funciones lambdas mínimo 3 por cada integrante del grupo.

- C. Implementar al menos 3 estructuras de datos distintas (listas enlazadas simples, dobles, circulares, pilas, colas), justificando el uso correcto, como mínimo una estructura de datos por cada integrante.
- D. Cada estructura de datos debe tener al menos 3 métodos implementados por cada integrante del grupo (diferente a los desarrollados en clase), por ejemplo, insertar, ordenar, buscar u otros.
- E. Aplicación de un algoritmo de ordenamiento avanzado por cada integrante.
- F. La aplicación debe ser presentada en consola, se debe considerar la usabilidad y estética de la aplicación.
- G. La calificación se realizará de manera individual en la exposición.

## 5. De la entrega

- El proyecto se realizará en grupos de 3 integrantes.
- La entrega consistirá únicamente de todos los archivos (.hpp y .cpp) o proyecto en visual Studio 2019/2022 que contenga los archivos usados (incluidos archivos de texto o binarios),
- Se debe realizar un video de grabación de la exposición de todos los integrantes no mayor a 5 minutos y un archivo Readme.txt. (Solo enviar el enlace del video)
- El archivo Readme.txt debe contener el detalle de desarrollo de cada uno de los integrantes del grupo con una autovaloración por el aporte y/o participación en una escala de 0-20.
- El nombre del archivo comprimido en formato zip y con los apellidos paternos de los integrantes del grupo, por ejemplo **TB1\_Codigo\_Apellido1\_Apellido2\_Nombre1\_nombre2.zip**
- La fecha de entrega es hasta el **11 de mayo de 2025 a las 23:59** por el aula virtual de blackboard.
- Adjuntar en un archivo de Microsoft Office Word donde muestre todos los procesos que se encuentran realizando, además, con una explicación del caso de estudio las estructuras necesarias y su complejidad en notación detallada, Big O del algoritmo que consideren principal por cada entidad e indicar los nombres de los integrantes y la descripción de tareas que realizó.

- El desarrollo del proyecto se realizará de manera grupal, pero la entrega es de manera individual.

## 6. De la presentación

- La primera sesión de clases de la semana 7, se hará una demostración y explicación del proyecto implementado, la elección para la exposición de los grupos y de los integrantes será aleatoria.
- El grupo puede exponer solamente si realizó el envío por el aula virtual de blackboard en la hora indicada.
- Todo avance posterior a la fecha de entrega no será considerado para la exposición.
- Todos los integrantes del grupo **deben conocer** en su totalidad la implementación.
- El estudiante debe presentar el T1 en el aula virtual de blackboard, de manera individual.
- El estudiante que no presenta el T1 no podrá exponer, donde tendrá nota CERO.
- El estudiante que presenta el T1 y no expone, tendrá nota CERO.
- Durante la exposición, el grupo debe declarar si ha utilizado herramientas de Inteligencia Artificial (IA) y debe especificar en qué parte del trabajo lo ha utilizado. En caso el grupo utilice una herramienta de IA y no lo declare, el docente tiene la potestad de penalizar si detecta el uso de estas herramientas. Asimismo, el docente tiene la prerrogativa de solicitar a cada integrante del grupo, que demuestre su conocimiento del código de programación que presenta. Si el estudiante no demuestra conocer su código de programación su calificación será penalizada.

## Rúbrica de calificación:

La Tabla 2 muestra la rúbrica de calificación por cada requisito que deben presentar en la aplicación

**La Tabla 2: Rúbrica de calificación**

<b>Criterio</b>	<b>Puntaje</b>
Diseñar el diagrama de clases UML	<b>3 puntos</b>
Análisis de Complejidad de 3 métodos que consideren más importantes.	<b>2 puntos</b>
Uso de lambdas y templates en cada estructura de datos justificados.	<b>2 puntos</b>
Implementación de estructuras de datos (listas simples, dobles enlazadas e iteradores, Pilas y Colas) con sus métodos respectivos	<b>6 puntos</b>
Implementación de al menos un método por estudiante con uso de recursividad.	<b>2 puntos</b>
Implementación de algoritmos de ordenamiento	<b>1 puntos</b>
Diseño de interfase amigable, usable en consola.	<b>2 puntos</b>
Uso de archivos de texto o binarios para leer y guardar registros, por cada una de las clases principales.	<b>2 puntos</b>

- Es imperativo el uso de POO y de Programación genérica en la implementación del proyecto.
- El puntaje se considera de manera individual en la exposición
- El acto de plagio con otros grupos es penalizado con la calificación de cero (0) para todos los integrantes involucrados.
- En el caso de obtener parte del código de otras fuentes de información en una sección de comentario referencie mediante la URL.

Lima, abril 2025